

# Evidence Estimation Using Stochastic Likelihood Approximations

by

Scott Cameron

*Thesis presented in fulfilment of the requirements for the  
degree of*

***Master of Science***

*(Physical and Mathematical Analysis)*



*in the Faculty of Science, Stellenbosch University*

Supervisor:

Prof H.C. Eggers

Department of Physics

Co-supervisor:

Prof S. Kroon

Computer Science Division

March 2020

The financial assistance of the National Institute of Theoretical Physics (NITheP) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to NITheP.

# Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date:                    March 2020  
.....

Copyright © 2020 Stellenbosch University  
All rights reserved.

# Abstract

## Evidence Estimation Using Stochastic Likelihood Approximations

S. Cameron

*Department of Physics,  
University of Stellenbosch,  
Private Bag X1, 7602 Matieland, South Africa.*

Thesis: MSc

March 2020

We consider the problem of estimating evidence for parametric Bayesian models in the large data regime. Many existing evidence estimation algorithms scale poorly due to their need to repeatedly calculate the exact likelihood, which requires iterating over the entire data set. This inefficiency can be circumvented with the use of stochastic likelihood estimates on small sub-samples of the data set. We therefore tackle this problem by introducing stochastic gradient Monte Carlo methods for evidence estimation, our main contribution being stochastic gradient annealed importance sampling. Our approach enables efficient online evidence estimation for large data sets. SGAIS is considerably faster than previous approaches for single data sets, with improved order complexity for online estimation, without noticeable loss of accuracy.

# Acknowledgements

I would like to thank the National Institute of Theoretical Physics for their generous funding towards my degree, as well as for a travel grant to attend MaxEnt 2019; the 39<sup>th</sup> International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering. Valuable discussions at the conference led to significant improvements to my work.

I would further like to thank the organizers of MaxEnt 2019 for generously sponsoring a research paper submitted to the journal Entropy.

# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Bayesian Evidence . . . . .	1
1.2 Online Estimation and Data Streaming . . . . .	4
1.3 Original Contributions . . . . .	5
1.4 Notation and Conventions . . . . .	6
1.5 Structure . . . . .	7
<b>2 Markov Chain Monte Carlo</b>	<b>9</b>
2.1 Metropolis–Hastings . . . . .	9
2.2 Hamiltonian Dynamics . . . . .	11
2.3 Hamiltonian Monte Carlo . . . . .	15
2.4 Langevin Dynamics . . . . .	18
2.5 Stochastic Gradient Markov Chain Monte Carlo . . . . .	24
<b>3 Algorithms For Evidence Estimation</b>	<b>30</b>
3.1 What Makes Evidence Estimation Difficult? . . . . .	31
3.2 Nested Sampling . . . . .	37
3.3 Annealed Importance Sampling . . . . .	43
3.4 Sequential Monte Carlo . . . . .	49
<b>4 Stochastic Gradient Evidence Estimation</b>	<b>54</b>
4.1 Sequential Evidence Estimation with SG-MCMC . . . . .	54
4.2 Stochastic Gradient Annealed Importance Sampling . . . . .	57

*CONTENTS***v**

4.3 Bayesian Updating with NS . . . . .	59
<b>5 Simulation Results</b>	<b>62</b>
5.1 Methodology . . . . .	62
5.2 Models . . . . .	65
5.3 Sequential Estimation with SG-MCMC . . . . .	66
5.4 Results For SGAIS . . . . .	69
<b>6 Conclusions</b>	<b>79</b>
6.1 Findings . . . . .	79
6.2 Limitations . . . . .	80
6.3 Future Work . . . . .	81
<b>List of References</b>	<b>82</b>

# List of Figures

2.1	Comparison of random walk MH to HMC . . . . .	18
2.2	Comparison of SGLD to SGHMC . . . . .	28
3.1	Prior and likelihood for a Gaussian–Gaussian model . . . . .	33
3.2	Likelihood weighted prior sampling . . . . .	34
3.3	Hamonic mean estimator histogram . . . . .	36
5.1	Non-stationary data . . . . .	64
5.2	Linear regression . . . . .	67
5.3	Logistic regression . . . . .	68
5.4	Gaussian mixture model . . . . .	69
5.5	Linear regression . . . . .	70
5.6	Logistic regression . . . . .	70
5.7	Gaussian mixture model . . . . .	71
5.8	Non-stationarity detection . . . . .	72
5.9	Sensitivity to number of particles . . . . .	73
5.10	Sensitivity to number of burn-in steps . . . . .	74
5.11	Sensitivity to mini-batch size . . . . .	75
5.12	Sensitivity to target ESS . . . . .	76
5.13	Sensitivity to learning rate . . . . .	77
5.14	Sensitivity to learning rate and number of steps . . . . .	78

# List of Algorithms

1	Random Walk Metropolis–Hastings . . . . .	11
2	Hamiltonian Monte Carlo . . . . .	17
3	Stochastic Gradient Langevin Dynamics . . . . .	27
4	Stochastic Gradient Hamiltonian Monte Carlo . . . . .	28
5	Nested Sampling . . . . .	42
6	Annealed Importance Sampling . . . . .	48
7	Sequential Importance Resampling . . . . .	53
8	Stochastic Gradient Annealed Importance Sampling . . . . .	60



# List of Abbreviations

<b>AIS</b>	Annealed importance sampling
<b>ESS</b>	Effective sample size
<b>HMC</b>	Hamiltonian Monte Carlo
<b>i.i.d.</b>	Independently and identically distributed
<b>MALA</b>	Metropolis adjusted Langevin algorithm
<b>MCMC</b>	Markov chain Monte Carlo
<b>MH</b>	Metropolis–Hastings
<b>NS</b>	Nested sampling
<b>ODE</b>	Ordinary differential equation
<b>SDE</b>	Stochastic differential equation
<b>SGAIS</b>	Stochastic gradient annealed importance sampling
<b>SGHMC</b>	Stochastic gradient Hamiltonian Monte Carlo
<b>SGLD</b>	Stochastic gradient Langevin Dynamics
<b>SG-MCMC</b>	Stochastic gradient Markov chain Monte Carlo
<b>SIR</b>	Sequential importance resampling
<b>SMC</b>	Sequential Monte Carlo

# Chapter 1

## Introduction

### 1.1 Bayesian Evidence

#### 1.1.1 Definition and Interpretation

Bayesian modeling presents the task of answering questions about data as posterior inference. Given observations  $\mathcal{D}$  and a model  $\mathcal{M}$  with parameters  $\theta$ , the posterior distribution is given by Bayes' rule

$$\underbrace{p(\theta|\mathcal{D}, \mathcal{M})}_{\text{posterior}} = \frac{\overbrace{p(\mathcal{D}|\theta, \mathcal{M})}^{\text{likelihood}} \overbrace{p(\theta|\mathcal{M})}^{\text{prior}}}{\underbrace{p(\mathcal{D}|\mathcal{M})}_{\text{evidence}}}. \quad (1.1)$$

Here  $\mathcal{M}$  is to be thought of as a model class. It could be a description of the model assumptions describing the data generating process. It is important, at this level, to be able to distinguish between the model class and its parameters. Different models may have different numbers of parameters and these will need to be marginalized out for fair model comparison.

The evidence (or marginal likelihood)  $\mathcal{Z} := p(\mathcal{D}|\mathcal{M})$  is the probability (or density) of the observations under the model assumptions. We adopt the convention that probabilities/densities are represented by the letter  $p$ , and distinguish between each probability distribution by the symbol in the argument. We do not further distinguish between probabilities and probability densities unless the context is ambiguous.

Many questions that need to be answered within the assumptions of the model usually only require the posterior distribution up to a normalizing constant,

$$p(\theta|\mathcal{D}, \mathcal{M}) \propto p(\mathcal{D}|\theta, \mathcal{M})p(\theta|\mathcal{M}), \quad (1.2)$$

and so the evidence is often neglected. However, if the model is unable to adequately describe the data, the posterior distribution may give ridiculous predictions since the model assumptions are taken to be ground truth; they appear only on the right hand side of the conditional. We can allow the model assumptions to be probabilistic in nature by specifying a prior probability distribution over a family of models  $\{\mathcal{M}_k\}_{k \in \mathcal{I}}$ . We cannot allow this family to be arbitrary, since it must be possible to construct a probability measure over it, so we typically only consider finitely many, or countably many models. The posterior distribution over this family

$$p(\mathcal{M}_k|\mathcal{D}) \propto \underbrace{p(\mathcal{D}|\mathcal{M}_k)}_{\text{model evidence}} p(\mathcal{M}_k), \quad (1.3)$$

is proportional to the model evidences. The model's evidences here play the sample role as likelihood values. For this reason, a model's evidence can be seen as a quantitative measure of how well that model describes the data set: the larger the evidence, the more likely it is that that model was the one which generated the data.

In practice, a uniform prior is often assumed. In this case the posterior distribution over models is given by a constant multiplied by the model evidences.

$$p(\mathcal{M}_k|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}_k). \quad (1.4)$$

By specifying a finite set of models, we are able to account for some model uncertainty; however our particular choice of models still encodes certain assumptions about the relationships between observations which may not be correct.

The specification of at least two competing models is essential to the iterative and perpetually provisional knowledge framework of science. No model is ever final, and no model is ever “absolutely correct”.

### 1.1.2 Model Combination and Selection

Calculation of the evidence  $p(\mathcal{D}|\mathcal{M})$  is crucial both in experimental science and in machine learning, but plays a different role in each. In science, we often have

some physically interpretable parameters in our models. These parameters could be the mass of a proton, the temperature of the sun, or the structure of molecules in cell walls. In these scenarios, we often want to find one model that best describes the observed phenomena. In these cases the evidence of a model is sometimes used as a selection criterion (Linden *et al.*, 2014, chapter 17; Barber, 2012, chapter 12); If one model has a significantly higher evidence than another, it is clearly a much better description of our observations. One nice feature of using evidence in this way, is that it automatically incorporates Occam's razor; models which are overly complex will have a lower evidence than those that are simpler but still sufficient to describe the data (Linden *et al.*, 2014, chapter 3).

In a machine learning context, on the other hand, the goal is not always to find the model with an optimal parameterization for the physical quantities. Rather, machine learning often seeks the best possible predictions for future observations  $y'$ , given present data  $\mathcal{D}$ . Given multiple models, we may therefore combine the predictions through a weighted sum of posterior predictive distributions over the models

$$p(y'|\mathcal{D}) = \sum_k p(y'|\mathcal{D}, \mathcal{M}_k) p(\mathcal{M}_k|\mathcal{D}). \quad (1.5)$$

The combination of models itself follows from elementary rules of probability theory. It weights the prediction  $p(y'|\mathcal{D}, \mathcal{M}_k)$  of each model by its ability, as quantified by the respective model posterior  $p(\mathcal{M}_k|\mathcal{D})$ , to describe the data already available.

Each model's individual posterior predictive  $p(y'|\mathcal{D}, \mathcal{M}_k)$  can in turn be written in terms of that model's posterior  $p(\theta_k|\mathcal{D}, \mathcal{M}_k)$  and the appropriate conditional distribution  $p(y'|\theta_k, \mathcal{D}, \mathcal{M}_k)$  for  $y'$ ,

$$p(y'|\mathcal{D}, \mathcal{M}_k) = \int p(y'|\theta_k, \mathcal{D}, \mathcal{M}_k) p(\theta_k|\mathcal{D}, \mathcal{M}_k) d\theta_k. \quad (1.6)$$

If the parameters are discrete, the integral is replaced by a sum.

The model posterior  $p(\mathcal{M}_k|\mathcal{D})$  in Equation (1.5) can be written in terms of the evidence  $p(\mathcal{D}|\mathcal{M}_k)$  by means of Bayes' rule,

$$p(\mathcal{M}_k|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{M}_k) p(\mathcal{M}_k)}{\sum_j p(\mathcal{D}|\mathcal{M}_j) p(\mathcal{M}_j)}. \quad (1.7)$$

Given equal and constant model priors, the combined model predictions can be written in terms of the individual predictions and model evidences  $\mathcal{Z}_k := p(\mathcal{D}|\mathcal{M}_k)$ .

$$p(y'|\mathcal{D}) = \frac{\sum_k p(y'|\mathcal{D}, \mathcal{M}_k) \mathcal{Z}_k}{\sum_k \mathcal{Z}_k}. \quad (1.8)$$

## 1.2 Online Estimation and Data Streaming

Many practical inference problems arise in situations where new data is continually made available. In these problems, the data may or may not be a time series with some inherent dynamical structure which should be modeled. Some examples of these kinds of data are stock prices, monthly weather data such as average temperatures and rainfall, and general user trends on websites.

In an online setting, recalculating the full evidence for every new batch of data, based on the all of the previous observations may be prohibitively inefficient. In data streaming applications, data may be arriving frequently enough that recomputing quantities from scratch every time could be slow enough to render the model useless if the data arrival rate exceeds the computation rate. It is therefore desirable to be able to calculate the evidence in a manner which efficiently updates previous estimates in such a way that the marginal time complexity is constant in the data set size. Processing of particle collision data at the Large Hadron Collider, or the filtering of radio frequency interference at the Square Kilometer Array are two examples of online applications with extreme speed requirements ([Brumfiel, 2011](#); [SKA, 2019](#))

In such online problems, inference may be formulated in terms of Bayesian updating. The initial data is used to update the prior to the first posterior distribution and when new data arrives, the previously calculated posterior is then treated as the prior giving a new posterior which incorporates all of the data so far. For a data set  $\mathcal{D} = \{y_n\}_{n=1}^N$ , and assuming the data are independent when conditioned on the model parameters, the  $n^{\text{th}}$  posterior can be calculated recursively

$$p(\theta|y_{<n+1}) = \frac{p(y_n|\theta)p(\theta|y_{<n})}{p(y_n|y_{<n})}, \quad (1.9)$$

with  $y_{<n}$  shorthand for the list  $(y_1, y_2, \dots, y_{n-1})$ . The denominator  $p(y_n|y_{<n})$ , which could be called the conditional evidence, is the posterior predictive distribution under the previous posterior  $p(\theta|y_{<n})$ .

In such online applications, the data may exhibit non-stationarities to an extent that the model is be unable to describe. For models which assume conditionally independent data this could be due to any change in the underlying data generating process. For models which explicitly model the time dynamics through some parameters  $\theta$ , such as autoregressive models, this can arise if the optimal values of  $\theta$  to describe the process at one period of time differ significantly from the optimal values of  $\theta$  to describe the process at a later time. Such changes can occur at discrete point in time, which could be modeled by change points, or they could occur continuously. When such extra-model non-stationarities arise, we would expect the evidence of the model to decrease, since it is not adequate to describe such behaviour. In this case the evidence can be used for change point detection or to assess the extent to which the model is able to capture the incremental non-stationarity of the data.

### 1.3 Original Contributions

This thesis aims to address, at least in part, the goal and technicalities of online evidence estimation and evidence estimation in the large data regime. The pertinent original work by the author, which forms the main contribution of this thesis, is set out in detail in subsequent chapters and summarised in Chapter 6. Briefly, the author's original contribution encompasses the following:

- We introduce and discuss approaches to estimating evidence using stochastic gradient Markov chain Monte Carlo methods and mini-batching.
- We introduce stochastic gradient annealed importance sampling (SGAIS), which combines stochastic gradient Markov chain Monte Carlo with annealed importance sampling to estimate the evidence in an online fashion using mini-batch Bayesian updating.
- By introducing SGAIS, we enable efficient evidence estimation for streaming data and for large data sets, which was not previously feasible.

- We illustrate how SGAIS can be used to identify distribution shifts in data when applied in an online setting.
- We empirically analyze the behavior of SGAIS and its robustness to various choices of algorithm parameters.

The original work appearing in this thesis has recently been published in [Cameron \*et al.\* \(2019a\)](#) and [Cameron \*et al.\* \(2019b\)](#).

## 1.4 Notation and Conventions

Throughout this thesis, we will denote parameters by  $\theta$ , observations by  $y$  or similar, a data set by  $\mathcal{D} = \{y_n\}_{n=1}^N$  and most probabilities by  $p$ . We sometimes write  $x$  for a generic variable without attaching the specific meaning of a parameter or observation. We will not differentiate between probabilities or probability densities but rather by their arguments, except when this convention may cause ambiguity.

We mainly consider parametric models with conditionally independent data. In these models the joint probability distribution factorizes into the prior times a product of individual-data likelihoods,

$$p(\mathcal{D}, \theta) = p(\theta) \prod_n p(y_n | \theta). \quad (1.10)$$

This restriction ensures that estimates of the log-likelihood using mini-batches sampled i.i.d. from the data set are unbiased, and are approximately normally distributed due to the central limit theorem. The restriction to conditionally independent data can be weakened to conditionally Markov data or autoregressive models since a log-likelihood of the form

$$\log p(\mathcal{D} | \theta) = \sum_n \log p(y_n | y_{n-1}, \dots, y_{n-k}, \theta) \quad (1.11)$$

would also yield to a central limit theorem for fixed  $k$ . This work is therefore applicable to many models in machine learning, with the notable exception of Gaussian processes. The reason for this is that the probability of a data set under a non-degenerate Gaussian process likelihood cannot be written in the above form for any fixed value  $k$ .

We will further assume that the parameters  $\theta$  are continuous variables and that the prior probability density function and the likelihood are continuous and differentiable. This allows us to use algorithms such as Hamiltonian Monte Carlo and stochastic gradient Hamiltonian Monte Carlo. Discrete parameters or hyper-parameters can be used in certain cases, in which case Gibbs sampling could be used to update them between the continuous parameter updates. In order for the mini-batching to still be beneficial, it would be required that the Gibbs update for the discrete parameters does not depend on the entire data set.

We do not use boldface for vectors as is the common convention, because many equations, such as those in Section 2.2, apply to more general structures such as Riemannian manifolds. Familiarity with differential equations and vector calculus is assumed, and we make use of differential operators in matrix equations. For example, given a matrix valued function  $A(x)$  and scalar function  $f(x)$ , we could denote a scalar function  $g(x) = \nabla^T A(x) \nabla f(x)$ , which would be interpreted as

$$g(x) = \sum_{j,k} \frac{\partial}{\partial x_j} A_{j,k}(x) \frac{\partial}{\partial x_k} f(x). \quad (1.12)$$

When there are multiple vector-valued variables involved, such as  $x$  and  $v$  then we may write  $\nabla_x$  and  $\nabla_v$  to mean the gradient with respect to  $x$  and gradient with respect to  $v$  respectively.  $\nabla_{x,v}$  would then mean gradient with respect to the vector  $r = (x, v)^T$ , the direct sum of  $x$  and  $v$ . For a vector-valued function, the divergence  $\nabla \cdot f(x)$  is the inner product of the differential operator and the vector function; in matrix notation it can equivalently be written  $\nabla^T f(x)$ . The Laplacian operator is defined to be  $\nabla^2 := \nabla \cdot \nabla$ .

## 1.5 Structure

Chapter 2 gives a brief introduction to Markov chain Monte Carlo. We start by discussing the basic theory along with the ubiquitous Metropolis–Hastings method. We then provide an outline of the theory behind the efficient Hamiltonian Monte Carlo algorithm, after which we give an introduction to stochastic gradient based Markov chain Monte Carlo, in particular stochastic gradient Hamiltonian Monte Carlo. Many evidence estimation algorithms require Markov chain Monte Carlo steps and so this material is covered first.



Chapter 3 covers some existing evidence estimation algorithms. We first give a description of the challenges which arise in evidence estimation and attempt to gain some small insight into the problem. This chapter then covers Nested Sampling, Annealed Importance Sampling and a very brief introduction to Sequential Monte Carlo, restricted to the scope of the models which we are considering.

Chapter 4 includes and largely reproduces our original work which has recently been published. We describe our approach to efficient, large-scale evidence estimation using stochastic gradient algorithms. The first approach, described in Section 4.1 was proposed in a conference paper contributed to *MaxEnt 2019* in July 2019. Subsequently, we extended our work to a research paper has been published in the peer-reviewed journal *Entropy*. The algorithm in [Cameron et al. \(2019b\)](#), which we called “stochastic gradient annealed importance sampling” is described in Section 4.2.

Chapter 5 describes the simulation experiments which we performed in our papers. We first describe the methodology and the models which we use to validate our approach followed by the results and discussion for the various experiments.

The Conclusions in Chapter 6 tie together all the issues and summarise what has been achieved and what remains to be done.

## Chapter 2

# Markov Chain Monte Carlo

All of the algorithms for evidence estimation presented in Chapter 3 will, at some stage, require Markov transitions. This chapter will cover some theory of Markov chain Monte Carlo (MCMC) and present an introduction to Hamiltonian Monte Carlo (HMC) which will set the stage for the introduction of stochastic gradient Markov chain Monte Carlo (SG-MCMC) algorithms.

Our main interest is in Bayesian modeling, so we almost exclusively use MCMC to generate parameter samples  $\theta_i$ ; however, MCMC algorithms are more generally applicable to simulation problems and so we will usually use the symbol  $x$  in this chapter to refer to the random variable which is being simulated.

### 2.1 Metropolis–Hastings

The basic principle of any Monte Carlo technique is to use random sampling to estimate quantities of interest described as expectations. Given any function  $h(x)$  of any variable  $x$ , the most basic Monte Carlo algorithm would estimate a quantity of interest

$$H := \mathbb{E}_{p(x)} [h(x)], \quad (2.1)$$

by generating  $M$  independent samples  $x_i$  directly from the probability distribution  $p(x)$  and calculating the estimator

$$\hat{H} := \frac{1}{M} \sum_{i=1}^M h(x_i), \quad x_i \stackrel{\text{i.i.d.}}{\sim} p(x). \quad (2.2)$$

Unfortunately, for many quantities of interest, the distribution  $p$  is difficult, or impossible to sample directly. A common example occurring in Bayesian inference is the estimation of posterior predictive distributions  $p(y'|\mathcal{D})$  by sampling from the posterior  $p(\theta|\mathcal{D})$ . Since the posterior distribution of interesting models is usually quite complex, generating samples  $\theta_i$  directly from it is not feasible. MCMC presents a practical solution to the problem of estimating expectations over complex distributions by simulating an ergodic<sup>1</sup> Markov chain which has  $p(x)$  as its stationary distribution (Brooks *et al.*, 2011, chapter 1). Assume a transition kernel  $k(x'|x)$  leaves  $p(x)$  invariant

$$\int k(x'|x)p(x) dx = p(x'), \quad (2.3)$$

then by the law of large numbers, averages taken over the realization of the Markov chain will converge to the expectation under the stationary distribution;

$$\frac{1}{M} \sum_{i=1}^M h(x_i) \xrightarrow{M \rightarrow \infty} \mathbb{E}_{p(x)} [h(x)], \quad (2.4)$$

when  $x_i$  is sampled from  $k$  based at  $x_{i-1}$

$$x_i \sim k(\cdot | x_{i-1}). \quad (2.5)$$

Many MCMC algorithms rely on detailed balance,  $k(x'|x)p(x) = k(x|x')p(x')$ , to ensure that they have the desired stationary distribution. One simple way to ensure this is to propose a new state  $x'$  by sampling from an arbitrary proposal distribution  $q(x'|x)$  which has unbounded support, and accept the new state with probability

$$\min \left\{ 1, \frac{p(x')q(x|x')}{p(x)q(x'|x)} \right\}, \quad (2.6)$$

and otherwise reject  $x'$  and keep  $x$  as the new state. This is called the Metropolis–Hastings (MH) rejection step. If the proposal distribution is symmetric then the acceptance probability reduces to  $\min\{1, p(x')/p(x)\}$ . One benefit of MH based algorithms is that they do not require knowledge of the normalization constant of the stationary distribution. This allows these algorithms to be used for Bayesian inference by just specifying the joint distribution  $p(\mathcal{D}, \theta) = p(\mathcal{D}|\theta)p(\theta)$ , and for equilibrium physics simulations by using

---

<sup>1</sup>The ergodicity is required to guarantee convergence.

the Boltzmann factor  $e^{-H/k_B T}$ . One of the simplest MCMC algorithms based on this is random-walk Metropolis–Hastings which uses a Gaussian proposal distribution centered around the current point  $x_{i-1}$ . Random walk Metropolis–Hastings as described in Algorithm 1 is easy to implement correctly and does not require any other user input so it can be useful for simple experiments.

---

**Algorithm 1** Random Walk Metropolis–Hastings

---

**Input:** stationary distribution  $p(x)$ , step size  $\epsilon$ , initial state  $x_0$

**Output:** samples  $x_{1:M} := \{x_1, x_2, \dots, x_M\}$

```

1: for  $i = 1, \dots, M$  do
2:   sample  $x' \sim \mathcal{N}(x_{i-1}, \epsilon)$ 
3:   sample  $u \sim U(0, 1)$  ▷  $U$  is the uniform distribution
4:   if  $u < \frac{p(x')}{p(x)}$  then ▷ The proposal distribution is symmetric
5:     set  $x_i \leftarrow x'$ 
6:   else
7:     set  $x_i \leftarrow x_{i-1}$ 
8:   end if
9: end for

```

---

## 2.2 Hamiltonian Dynamics

The random walk proposal distribution of Algorithm 1 results in slow convergence to the stationary distribution due to high correlations between successive samples  $x_i$ . Furthermore, this high autocorrelation implies that each new sample contains little new information, even after the chain has converged to the desired stationary distribution after the so-called burn-in phase. In order to combat this, more sophisticated MCMC kernels need to be introduced. This section will attempt to shed some light on Hamiltonian dynamics with the purpose of using Hamiltonian simulations as MCMC proposals. A reader with a strong background in physics may safely skip this section and move directly to Section 2.3

The formalism of Hamiltonian dynamics describes the physics of classical objects. Hamiltonian dynamics is equivalent to Newtonian mechanics if there are no dissipative forces; however, it is a far more elegant construction, naturally describing conservation laws, in more general coordinate systems than just Euclidean. The full machinery of Hamiltonian dynamics and its construction in

classical mechanics is beyond the scope of this thesis; we will merely attempt to describe the basics of Hamiltonian dynamics and highlight some important properties. See [Arnold \(1989\)](#) for a rigorous mathematical introduction to the topic.

We may parameterize the configuration of a classical system by coordinates  $x_i$ . Classical systems have associated conjugate momenta, which we will denote here as  $v_i$ . These are not necessarily velocities, however momenta are dual to velocities and so this notation is not completely inappropriate. We choose  $x$  and  $v$  here instead of the more conventional  $q$  and  $p$  in order to avoid confusion with probabilities. Together, the coordinates and momenta uniquely and completely specify the state of the system. The joint space of coordinates and momenta is called the phase space.<sup>2</sup> Classical systems have an associated Hamiltonian function  $H(\{x_i\}, \{v_i\})$  which is the generator of the dynamics of the system. Hamilton's equations of motion describe the system's trajectory through phase space given some initial conditions. Hamilton's equations of motion are

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial v_i} \quad \text{and} \quad \frac{dv_i}{dt} = -\frac{\partial H}{\partial x_i}. \quad (2.7)$$

The Hamiltonian is the total energy of the system, which can often be described as the sum of kinetic and potential energies which depend solely on  $x$  and on  $v$  respectively,

$$H(x, v, t) = K(v) + U(x, t). \quad (2.8)$$

For the usual quadratic kinetic energy  $K(v) = \frac{1}{2m}v^2$ , these equations of motion reduce to Newton's second law

$$m \frac{d^2x}{dt^2} = -\nabla U(x, t). \quad (2.9)$$

If the potential is time-independent,  $U(x, t) = U(x)$ , the Hamiltonian does not depend explicitly on time and thereby becomes a constant of motion,

$$\frac{dH}{dt} = \sum_i \left( \frac{\partial H}{\partial x_i} \frac{dx_i}{dt} + \frac{\partial H}{\partial v_i} \frac{dv_i}{dt} \right) + \frac{\partial H}{\partial t} \quad (2.10)$$

$$= \sum_i \left( \frac{\partial H}{\partial x_i} \frac{\partial H}{\partial v_i} - \frac{\partial H}{\partial v_i} \frac{\partial H}{\partial x_i} \right) + 0 \quad (2.11)$$

$$= 0. \quad (2.12)$$

---

<sup>2</sup>The phase space is the cotangent bundle of the configuration manifold.

This is the first important property of Hamiltonian dynamics. For use in MCMC, exact conservation in time of the Hamiltonian will result in a Metropolis–Hastings acceptance probability of one. Since dynamics are simulated numerically, the Hamiltonian will not be conserved exactly; however, the discrepancy will typically be small.

A second property of Hamiltonian dynamics is that it conserves volumes in phase space. The conservation of volume means that the Jacobian determinant of transforming a phase space point through Hamiltonian dynamics for any amount of time is one. This property, known as *Liouville’s theorem* in the context of Hamiltonian dynamics ([Arnold, 1989](#), chapter 3), is a fundamental theorem in classical mechanics and statistical physics. In the context of the Metropolis–Hastings algorithm it is also an important property. Since Hamiltonian dynamics is deterministic, the conditional probability of moving from one phase space configuration to another is given by a Dirac delta function, multiplied with a Jacobian factor. Since we already know that the Jacobian is unity, this greatly simplifies calculation of the MH acceptance ratio.

Thirdly, if the kinetic energy is symmetric,  $K(v) = K(-v)$ , then the dynamics can be reversed just by negating the momentum. Assuming that the system is initially in state  $(x, v)$ , and evolves under Hamilton’s equations of motion for some time to arrive at  $(x', v')$ , then it can be shown that starting with initial condition  $(x', -v')$  and evolving the system for the same amount of time, the system would arrive at  $(x, -v)$ . This can be seen by noting that a simultaneous time reversal  $t \rightarrow -t$  and parity transformation  $v \rightarrow -v$  leaves Equation 2.7 invariant.

Hamilton’s equations of motion define a flow through phase space which transports configurations of the system continuously. We may imagine an ensemble of systems or, phrased differently, a distribution over possible initial conditions, which follow these dynamics. We would expect to be able to calculate the dynamics of the distribution of this ensemble of systems over time from these equations of motion. It turns out that the distribution of configurations of a system whose dynamics are given by an ordinary differential equation follows a partial differential equation called *Liouville’s equation*, which is closely related to Liouville’s theorem. We will not give a formal proof here, but describe the form of the equation intuitively as follows. Consider a system which follows

dynamics governed by a first-order ordinary differential equation in terms of a velocity vector field  $b(x, t)$ ,

$$\frac{dx}{dt} = b(x, t). \quad (2.13)$$

Higher-order ODEs can be written in this form by transforming them into coupled first-order ODEs. Let  $p(x; t)$  be a distribution over configurations of the system at time  $t$ . This can be thought of as a particle density if the particles are not interacting, or as a probability density in  $x$  describing our uncertainty over the current configuration. There is a probability/particle *current* which is the product of the density and the velocity field

$$j := p(x; t) \frac{dx}{dt} = p(x; t) b(x, t). \quad (2.14)$$

The current is a vector field which gives the net density and direction of the flow of probability or particles away from each point in the space. The density and current naturally satisfy a continuity equation

$$\frac{\partial p}{\partial t} = -\nabla \cdot j = -\nabla \cdot (p b). \quad (2.15)$$

This continuity equation is Liouville's equation. Intuitively, it states that the rate at which the density increases at any point is equal to the net rate of particles entering that point minus the rate of particles leaving that point. Continuity equations arise whenever systems exhibit continuous motion or continuous symmetries. They are a kind of conservation law, in this case conservation of number of particles or conservation of total probability. Noether's theorem usually manifests itself as a continuity equation. For Hamilton's equations of motions, by substituting Equation 2.7 into Equation 2.15, one arrives at Liouville's equation for Hamiltonian dynamics

$$\frac{\partial p}{\partial t} = \nabla_v \cdot (p \nabla_x H) - \nabla_x \cdot (p \nabla_v H). \quad (2.16)$$

It is easily shown by simple substitution that it yields a family of stationary solutions in the form of a *Boltzmann distribution*

$$p(x, v) = \frac{1}{Z} e^{-\beta H(x, v)} \quad (2.17)$$

where the normalization constant  $Z$  is called the partition function. The positive constant  $\beta$  is a "Lagrange multiplier" whose value depends on the system constraints.

In physics,  $\beta$  is interpreted as an inverse temperature and the ensemble of points  $(x, v)$  governed by Hamilton's equations and the Boltzmann distribution is known as the *canonical ensemble*. In the context of MCMC, these need not have a physical interpretation at all, however it is still a useful analogy.

Since Hamiltonian dynamics conserves the Hamiltonian, any initial energy distribution is also stationary, and so closed Hamiltonian systems do not necessarily converge to the canonical ensemble. However the system can converge to the canonical ensemble if energy is allowed to enter and exit the system through a so called "heat bath". If the energy entering and exiting the system is correctly controlled then the system may exhibit a unique stationary distribution with a well defined temperature.

For the separable Hamiltonian  $H = K(v) + U(x)$ , the positions and momenta become statistically independent with marginals given by

$$p(x) = \frac{1}{Z_x} e^{-\beta U(x)}, \quad \text{and} \quad p(v) = \frac{1}{Z_v} e^{-\beta K(v)}. \quad (2.18)$$

## 2.3 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC), also called Hybrid Monte Carlo, is an MCMC algorithm for the variables  $x$  which introduces "momenta"  $v$  as auxiliary variables and uses simulations of Hamiltonian dynamics to propose new states in the Markov chain (Brooks *et al.*, 2011, chapter 5). HMC is a highly regarded MCMC algorithms due to its fast convergence and scalability to large and complex models. HMC asymptotically samples from the Boltzmann distribution with  $\beta = 1$ . One can specify a target distribution  $p(x)$  by choosing a Hamiltonian with potential energy  $U(x) = -\log p(x)$ .

$$H = -\log p(x) + K(v) + \text{const}. \quad (2.19)$$

Usually the kinetic energy is a quadratic form  $K(v) = \frac{1}{2}v^T M^{-1}v$  for some positive definite, symmetric matrix  $M$ , called the mass matrix.

Given a point  $x$ , one could sample a velocity from the appropriate canonical ensemble distribution  $e^{-K(v)}$  and simulate Hamiltonian dynamics with the



above Hamiltonian, arriving at  $x'$  and then discarding the final velocity. The resulting Markov kernel taking  $x$  to  $x'$  would leave the distribution  $p(x)$  invariant provided dynamics simulation is exact. By sampling the momenta at the beginning of each Markov transition and discarding them at the end, we effectively marginalize them out. Resampling momenta in this way is also an effective method of adding or removing energy from the system. This is essentially the same as coupling the system to a heat bath with unit temperature;  $\beta = 1$ . Allowing the energy to change in this way ensures that the Markov chain generated by simulating Hamiltonian dynamics is ergodic and guarantees convergence to the Boltzmann distribution with temperature one.

In general, Hamiltonian dynamics cannot be simulated exactly and so numerical integrators are used instead. This results in discretization errors and thereby does not result in the correct stationary distribution. While small step sizes help reduce the discretization error, Metropolis-Hastings corrections have to be introduced to guarantee convergence. To calculate the acceptance probability in Equation 2.6, we would normally need to know how to calculate the proposal density  $q(x'|x)$ . Fortunately, since Hamiltonian dynamics is reversible and has a Jacobian determinant of one, the proposal density is symmetric and so can be neglected in the MH rejection step. To capitalize on these properties, it is therefore necessary that the numerical integration preserves these two properties of the dynamics. These properties can be maintained by using the “leapfrog” integration algorithm. Leapfrog integration with step-size  $\epsilon$  proceeds as follows:

$$v_{t+\frac{1}{2}} = v_t - \frac{\epsilon}{2} \nabla U(x_t), \quad (2.20)$$

$$x_{t+1} = x_t + \epsilon \nabla K(v_{t+\frac{1}{2}}), \quad (2.21)$$

$$v_{t+1} = v_{t+\frac{1}{2}} - \frac{\epsilon}{2} \nabla U(x_{t+1}). \quad (2.22)$$

With this integrator, the MH acceptance probability, Equation 2.6, reduces to

$$\min \left\{ 1, \frac{\exp[-U(x') - K(v')]}{\exp[-U(x) - K(v)]} \right\}. \quad (2.23)$$

For small step-sizes, the energy will be approximately conserved and will result in very few rejections.

HMC is outlined in Algorithm 2 below with a quadratic kinetic energy and an identity mass matrix.

---

**Algorithm 2** Hamiltonian Monte Carlo
 

---

**Input:** potential energy  $U(x)$ , step size  $\epsilon$ , number of leapfrog steps  $L$ , initial state  $x_0$

**Output:** samples  $x_{1:M}$

```

1: for  $i = 1, \dots, M$  do
2:   sample  $v \sim \mathcal{N}(0, 1)$ 
3:   set  $x' \leftarrow x_{i-1}$ 
4:   set  $v' \leftarrow v - \frac{\epsilon}{2} \nabla U(x')$  ▷ half step
5:   for  $t = 1, \dots, L$  do
6:     set  $x' \leftarrow x' + \epsilon v'$ 
7:     if  $t < L$  then
8:       set  $v' \leftarrow v' - \epsilon \nabla U(x')$ 
9:     end if
10:  end for
11:  set  $v' \leftarrow v' - \frac{\epsilon}{2} \nabla U(x')$  ▷ half step
12:  sample  $u \sim U(0, 1)$ 
13:  if  $\log(u) < H(x_{i-1}, v) - H(x', v')$  then ▷ HM-accept/reject
14:    set  $x_i \leftarrow x'$ 
15:  else
16:    set  $x_i \leftarrow x_{i-1}$ 
17:  end if
18: end for
    
```

---

HMC has much faster convergence and much shorter autocorrelation time than random walk based MH algorithms. See Figure 2.1 for a comparison of trajectories generated by random walk MH and HMC. Introducing the auxiliary momenta allows the particle to travel longer distances, but also allows the particle to move to areas of lower probability without rejections simply by transforming potential energy into kinetic energy. The introduction of momenta is the crucial feature which allows the particle to better explore the space.

There are however some downsides to HMC, at least in its vanilla form with an isotropic quadratic kinetic energy. Firstly: If there are strong correlations between variables in the distribution of interest, then the algorithm would benefit by sampling large momenta in directions with large variance and smaller momenta in directions with low variance. Secondly: For complex distributions, there might be higher order correlations and changes in curvature along valleys in the potential energy, and so being able to vary the mass matrix depending on the position of the particle can improve equilibration and lower autocorrela-

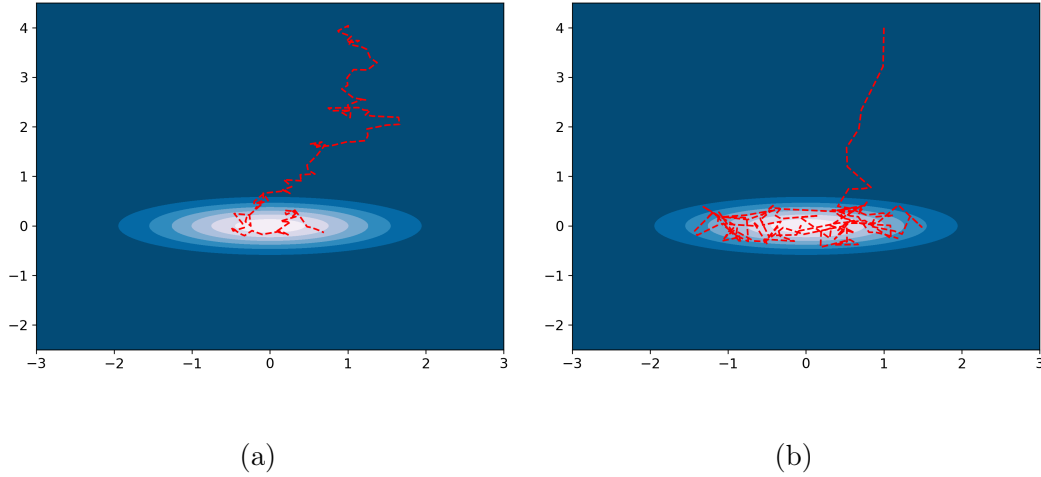


Figure 2.1: (a) shows a trajectory generated by random walk Metropolis–Hastings (Algorithm 1) and (b) shows a trajectory generated by Hamiltonian Monte Carlo (Algorithm 2).

tions. Riemannian manifold HMC (Girolami and Calderhead, 2011) addresses this by using the local curvature of the distribution as the covariance when sampling momenta. This requires some careful analysis since the kinetic energy will then depend on position. Various other extensions to HMC exist to improve convergence or mixing, or algorithmic efficiency. See Brooks *et al.* (2011, chapter 5) for more detail and discussion regarding HMC.

When using HMC for Bayesian inference, we usually want to generate samples from the posterior, in which case the potential energy is a sum over the parameter likelihood and prior,

$$U(\theta) = -\log p(\mathcal{D}|\theta) - \log p(\theta). \quad (2.24)$$

Calculating the potential energy and its gradients is at least linear time complexity in the data set size  $\mathcal{O}(|\mathcal{D}|)$ . This can be quite prohibitive for Bayesian inference on large data sets. One solution to this is to use stochastic gradient based MCMC algorithms, which we will introduce in Section 2.5.

## 2.4 Langevin Dynamics

The main difficulty with scaling Bayesian inference to large data sets using HMC is the requirement of iterating through the whole data set every time the potential energy, or its gradient, is required. Maximum likelihood estimation

and variational inference are still feasible on large data sets with the help of stochastic optimization. [Robbins and Monro \(1951\)](#) presented a method for finding roots of a function given only noisy, but unbiased, estimates of the function values. Under certain conditions the algorithm can be guaranteed to converge. One of these conditions is that the step sizes  $\epsilon_t$  follow a sequence which satisfies

$$\sum_{t=1}^{\infty} \epsilon_t = \infty, \quad \text{and} \quad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty. \quad (2.25)$$

Informally the first equation states that the total simulation time should be infinite, and therefore the root can be reached, no matter how far away from the initial point, while the second equation states that the total variance due to the noisy estimates (which are multiplied by the step sizes) should be finite. When such a step size sequence is applied to gradient descent or ascent, which corresponds to finding roots of the gradient, with an unbiased estimator of the gradients, then the algorithm will converge almost surely to a local optimum. This means that maximizing the likelihood, or maximizing the evidence lower bound in variational inference, can be done efficiently by performing gradient ascent using small sub-samples (mini-batches) of the data set instead of iterating over the entire data set. Stochastic gradient descent on a loss function using mini-batch gradient estimates is valid when the mini-batch estimates are unbiased. Using mini-batches like this can greatly reduce the convergence time of the algorithms.

Unfortunately naively using stochastic likelihood estimates in HMC violates the basic assumptions of the theory: If the potential energy function changes over time (whether stochastically or deterministically) then the Hamiltonian is no longer conserved. This can result in samples from a distribution which is very different to the Boltzmann distribution. See [Chen \*et al.\* \(2014\)](#) for an example of this. In order to introduce stochastic gradients in HMC, we must first consider how stochastic forces influence a classical physical system.

Complementing the usual deterministic forces in a classical system with stochastic forces results in what is known as Langevin dynamics ([Kampen, 2007](#), chapter 9). The most fundamental and simplest such system is the Brownian particle. The  $d$ -dimensional Brownian motion, or Wiener process ([Mackevicius, 2013](#), chapter 2), is a time-homogeneous Markov process with finite-time

transition kernel

$$p(x_{t+\tau}|x_t; t, \tau) = \frac{1}{(\sqrt{2\pi\tau})^d} \exp\left(-\frac{(x_{t+\tau} - x_t)^2}{2\tau}\right), \quad (2.26)$$

which is the solution to an isotropic, translation invariant diffusion equation

$$\frac{\partial p}{\partial t} = \frac{1}{2} \nabla^2 p \quad (2.27)$$

with initial condition  $p(x'|x; t, \tau = 0) = \delta(x' - x)$ . The diffusion equation is again a kind of continuity equation with probability current  $j = -\frac{1}{2}\nabla p$ . This continuity equation arises because Brownian motion is almost surely continuous. This current can be interpreted with the following argument. Imagine an infinitesimal (hyper-)cube. Particles flow isotropically in every direction, and therefore any particle on the surface of the cube has exactly one half probability of entering the cube, and one half probability of leaving the cube. The net flow of particles through the surface will be one half times the difference in number of particles on the outside surface and the number of particles on the inside surface. By taking the volume to be infinitesimally small, the net density of particles flowing through the volume will tend to exactly half of the gradient of the particle density, in the direction of steepest descent.

If the motion is not isotropic or homogeneous, then we may locally transform to a coordinate system in which the motion is isotropic, calculate the current in that coordinate system and then transform back. By doing the reverse transformation, the current picks up two Jacobian factors, one for the density and one for the gradient, the outer product of which is a symmetric positive definite matrix which becomes the *diffusion matrix*. Letting the diffusion matrix absorb the factor  $\frac{1}{2}$ , the resulting probability current is  $j = -\nabla(p(x) D(x))$ , where it is implied that the differential operator matrix multiplies on the right with the diffusion matrix. In other words, the components of the current are  $j_i = -\sum_k \frac{\partial}{\partial x_k} p(x) D_{i,k}(x)$ . This is the Fokker–Planck diffusion probability current and takes the interpretation that the inhomogeneity affects the process locally in a certain way. This is in contrast to Fick’s first law ([Gorban et al., 2010](#)) for diffusion in inhomogeneous media which gives a probability current  $j = -D(x)\nabla p(x)$ . Which probability current to use depends on the microscopic details of the diffusion.

The Wiener process is ubiquitous in science and has many interesting properties. For example, paths which are realizations of the Wiener process are

everywhere continuous with probability one, but they are nowhere differentiable. As such, the derivative of a Wiener process, which is continuous-time white noise, is not well defined in the sense of functions; however, much like the derivative of the Dirac delta, it can be defined as a linear functional and treated as a density due to the Riesz–Markov–Kakutani representation theorem (Rudin, 1987, chapters 2 and 6). The Wiener process is also self-similar, and exhibits the scaling law  $W(\alpha t) \cong \sqrt{\alpha} W(t)$ .<sup>3</sup> Scaling laws of this form are often studied in statistical physics and tend to only exist for simple functions which are power laws or for very complex non-differentiable functions such as the Wiener process or the Weierstrass function. This kind of self-similarity gives rise to fractal behaviour and has interesting implications in the study of complex systems, particularly near phase changes.

Langevin dynamics<sup>4</sup> describe classical particles in a medium by extending the forces found in Hamiltonian dynamics to include a dissipative friction force, due to the medium, and a stochastic force, which represents the individual molecules in the medium bumping the larger particle. The stochastic force is assumed to be white noise because the time-scales of the movement of the particle are typically much larger than the time-scales of its interaction with the molecules in the medium. In the informal notation used by physicists, we may write the second order Langevin equation with unit mass as

$$\frac{d^2x}{dt^2} = \underbrace{-\nabla U(x)}_{\text{conservative force}} - \underbrace{\gamma(x)\frac{dx}{dt}}_{\text{friction}} + \underbrace{\sqrt{2D(x)}\xi(t)}_{\text{stochastic force}}, \quad (2.28)$$

where  $\gamma(x)$  is the friction coefficient which is either a positive scalar or a positive semi-definite matrix, and  $D(x)$  is a symmetric positive semi-definite diffusion matrix; the square root is the local coordinate transformation whose inverse diagonalizes the stochastic force. The random variable  $\xi(t)$  follows a white noise distribution

$$\mathbb{E}[\xi(t)] = 0, \quad \text{and} \quad \mathbb{E}[\xi(t')\xi(t)] = \delta(t' - t). \quad (2.29)$$

---

<sup>3</sup>The congruency here means that this transformation is an isomorphism; in this case isomorphism would mean distribution preserving.

<sup>4</sup>Here we describe Langevin dynamics with a potential energy. Sometimes the term ‘Langevin dynamics’ is used to refer to the case where the potential is zero, and it is usually implied to be the first order variant when not otherwise specified.

If the medium is homogeneous then  $\gamma$  and  $D$  will be constants and, unless there is some external influence on the medium such as electromagnetic forces, the system will be isotropic and so  $\gamma$  and  $D$  will be scalars. If the system is over-damped ( $\gamma \rightarrow \infty$ ), the effective dynamics reduces to the first order Langevin equation

$$\frac{dx}{dt} = -\nabla U(x) + \sqrt{2D(x)} \xi(t). \quad (2.30)$$

The diffusion matrix here is not necessarily equal to the diffusion matrix appearing in the second order equation. The typical behaviour of these dynamics is to initially descend to the minimum of the potential energy, and then slowly diffuse around near the minimum.

A more formal treatment relies on the theory of stochastic differential equations (SDEs). The Langevin equations can be written somewhat more formally as SDEs in the Itô<sup>5</sup> interpretation where the driving stochastic process is a Wiener process. Expectations of quantities depending on the realization of the SDE can be written as Wiener integrals, in which case they are usually calculated perturbatively. As before with Liouville's equation and the diffusion equation, we can write down a partial differential equation for the probability density based on the probability current. For an Itô SDE driven by a Wiener process,

$$dx = \underbrace{\mu(x, t) dt}_{\text{drift}} + \underbrace{\sigma(x, t) dW(t)}_{\text{diffusion}}, \quad (2.31)$$

the probability density  $p(x; t)$  evolves according to the Fokker–Planck equation

$$\frac{\partial p}{\partial t} = \underbrace{-\nabla \cdot (p \mu)}_{\text{drift}} + \overbrace{\sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j} (p D_{i,j})}^{\text{diffusion}}, \quad (2.32)$$

where the diffusion matrix is  $D = \frac{1}{2} \sigma \sigma^T$ . The Fokker–Planck equation is also sometimes called the (forward) Kolmogorov equation, although the Kolmogorov equation may also contain terms corresponding to jump processes. If the driving stochastic process is not Gaussian, then there exist generalizations to the above equation involving higher order derivatives. Note how the drift term in the Fokker–Planck equation corresponds to Liouville's equation for

---

<sup>5</sup>Unlike Riemann integration, stochastic integration depends strongly on how the discretization is done. The two most prominent variants are the Itô and Stratonovich formulations (Mackevicius, 2013, chapters 7 and 8)

the ODE that results in Equation 2.31 by setting  $\sigma$  to zero, and the diffusion term corresponds to the diffusion equation which results when there is no drift velocity. The probability current can be written exactly as the sum of the drift current and the (not necessarily homogeneous or isotropic) diffusion current  $j = j_{\text{drift}} + j_{\text{diff}}$ .

For homogeneous and isotropic diffusion ( $D$  is a scalar constant), first order Langevin dynamics in a potential written as an Itô SDE is

$$dx = -\nabla U(x)dt + \sqrt{2D} dW(t), \quad (2.33)$$

with the corresponding Fokker–Planck equation

$$\frac{\partial p}{\partial t} = \nabla \cdot (p \nabla U(x)) + D \nabla^2 p. \quad (2.34)$$

The Fokker–Planck equation for first order Langevin dynamics admits a unique stationary solution

$$p(x) = \frac{1}{Z} e^{-\beta U(x)}, \quad (2.35)$$

where the temperature is given by the diffusion constant  $\beta = 1/D$ . This stationary distribution is exactly the position marginal of the canonical ensemble. One interesting consequence of this is that the temperature of the particles can be controlled by increasing or decreasing the magnitude of the stochastic force. At equilibrium, the temperature of the medium is the same as the temperature of the Brownian particles, so the variance of the stochastic force can be directly interpreted as the temperature of the medium.

For second order Langevin dynamics, we will assume that the diffusion matrix does not depend on velocity, but may depend on position and that it is a constant scalar multiple of the friction. This assumption is for mathematical convenience. This is trivially the case for the most common scenario where both the diffusion and friction coefficients are constant and scalars. The Itô SDE is

$$dx = v dt, \quad (2.36)$$

$$dv = -\nabla U(x) dt - \gamma(x)v dt + \sqrt{2\gamma(x)T} dW(t), \quad (2.37)$$

where  $T$  is the multiplicative constant. The corresponding Fokker–Planck equation can be written compactly in matrix form (Chen *et al.*, 2014)

$$\frac{\partial p}{\partial t} = \nabla_{x,v}^T \{ [A + B]p \nabla_{x,v} H + B T \nabla_{x,v} p \} \quad (2.38)$$



where  $H = U(x) + \frac{1}{2}v^2$ , and  $A$  and  $B$  are the matrices

$$A = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & \gamma(x) \end{bmatrix}. \quad (2.39)$$

Noting that  $\nabla_{x,v}^T A \nabla_{x,v} p = \nabla_x^T \nabla_v p - \nabla_v^T \nabla_x p = 0$ , the Fokker–Planck equation can be written equivalently as

$$\frac{\partial p}{\partial t} = \nabla_{x,v}^T [A + B] \{p \nabla_{x,v} H + T \nabla_{x,v} p\}. \quad (2.40)$$

This equation again admits the Boltzmann distribution as the unique stationary solution

$$p(x, v) = \frac{1}{Z} e^{-\beta H(x, v)}, \quad (2.41)$$

where  $\beta = 1/T = \gamma(x)/D(x)$ .

Similarly to Hamiltonian dynamics, Langevin dynamics exhibits a kind of reversibility. This reversibility is of the form

$$p(x', v' | x, v; t) = p(x, -v | x', -v'; t). \quad (2.42)$$

This can be shown by viewing the right hand side of the Fokker–Planck equation as an operator acting on a Hilbert space and finding the adjoint operator ([Chen et al., 2014](#)).

## 2.5 Stochastic Gradient Markov Chain Monte Carlo

Just as HMC is inspired by Hamiltonian dynamics, we can develop stochastic gradient MCMC (SG-MCMC) algorithms taking inspiration from Langevin dynamics. Both first order and second order Langevin dynamics admit the Boltzmann distribution as a stationary distribution under certain conditions, so we can simulate either dynamics to produce samples from a desired distribution by setting  $U(x) = -\log p(x)$ . First order dynamics will then converge to  $p(x)$  if a diffusion constant of one is used, and second order dynamics will converge to  $p(x)$  if the diffusion matrix is equal to the friction coefficient. We can use SG-MCMC algorithms for efficient large-scale Bayesian inference by

estimating the potential energy using mini-batches sampled uniformly from the data set

$$\hat{U}(\theta) := -\frac{|\mathcal{D}|}{|B|} \sum_{y \in B} \log p(y|\theta) - \log p(\theta). \quad (2.43)$$

with  $|\mathcal{D}|$  and  $|B|$  the number of samples in the entire data set and the batch respectively. For mini-batches of sufficient size, the gradients of the potential energy estimate will be approximately Gaussian distributed.

As with HMC, we need an integrator to simulate the dynamics. HMC relies on leapfrog integration to preserve certain favourable properties and on MH rejection steps to correct for discretization error. Small step sizes for HMC result in fewer rejections, but result in more likelihood gradient calculations for the same distance travelled. In this case the MH corrections still guarantee convergence even for large step sizes. For SG-MCMC algorithms, we can use the Euler–Maruyama integration scheme (Mackevicius, 2013, chapter 13). For an SDE of the form

$$dx = \mu(x) dt + \sigma(x) dW(t), \quad (2.44)$$

the Euler–Maruyama integrator with step-size  $\epsilon$  performs the update rule

$$x_t = x_{t-1} + \mu(x_{t-1}) \epsilon + \sigma(x_{t-1}) \xi_t, \quad (2.45)$$

where  $\xi_t$  is a normally distributed random vector with variance  $\epsilon$ . Discretization error will still be a concern, however we can control for it by carefully decreasing the step size. With the use of mini-batches, estimating likelihood gradients is computationally inexpensive in comparison to an MH correction which requires iterating over the entire data set. Therefore using a very small step size does not result in computational inefficiency as it would with HMC.

Euler–Maruyama integration with step size  $\eta$  for first order Langevin dynamics has the following update rule

$$x_t = x_{t-1} - \eta \nabla U(x_{t-1}) + \xi_t, \quad \xi_t \sim \mathcal{N}(0, 2\eta). \quad (2.46)$$

This equation is the same as the update rule for (stochastic) gradient descent, except for the added noise  $\xi_t$ . Therefore, following common practice in machine learning, we call  $\eta$  the learning rate. As it turns out, performing a single leapfrog step of HMC, without an MH correction is equivalent to the following

update step

$$x_t = x_{t-1} - \frac{\epsilon^2}{2} \nabla U(x_{t-1}) + \epsilon v, \quad v \sim \mathcal{N}(0, 1). \quad (2.47)$$

These updates are identical and we can identify  $\eta = \frac{1}{2}\epsilon^2$ . For this reason HMC with a single leapfrog step is also called the Metropolis adjusted Langevin algorithm (MALA).

Although we can calculate the MH acceptance probability based on a single leapfrog step if we have access to the exact gradients, it is no longer possible if we use stochastic approximations of the gradients based on mini-batches, since we can no longer calculate the probability of the reverse transition. The variance of the gradient term is proportional to  $\eta^2$ , while the variance of the injection noise  $\xi$  is proportional to  $\eta$ . Therefore using a small learning rate results in the gradient noise being dominated by the injection noise, and so it can typically be ignored for a small enough learning rate. We can also account for the gradient noise by estimating its variance and subtracting that estimate from the variance of the injection noise.

To guarantee convergence we can use a learning rate schedule following the Robbins–Monro conditions ([Welling and Teh, 2011](#))

$$\sum_{t=1}^{\infty} \eta_t = \infty, \quad \text{and} \quad \sum_{t=1}^{\infty} \eta_t^2 < \infty; \quad (2.48)$$

however, using a small learning rate  $\eta \sim \mathcal{O}\left(\frac{1}{|\mathcal{D}|}\right)$  is usually sufficient in practice.

Simulating first order Langevin dynamics with stochastic gradient estimates is called stochastic gradient Langevin dynamics (SGLD) and was introduced by [Welling and Teh \(2011\)](#). SGLD is summarized in Algorithm 3. The parameter  $\hat{\beta}$  in Algorithm 3 is to control for the stochastic gradient variance. It is meant to be an estimate of the variance of the stochastic gradients, which may be calculated adaptively or user-specified.

One potential downside of SGLD is that it no longer uses the auxiliary momenta which were introduced in HMC, because it is based on first order Langevin dynamics, which corresponds to the infinite friction limit. While following gradients converges far quicker than a simple random walk, SGLD

---

**Algorithm 3** Stochastic Gradient Langevin Dynamics
 

---

**Input:** unbiased potential energy estimate  $\hat{U}(x)$ , learning rate  $\eta$ , noise estimate  $\hat{\beta}$ , initial state  $x_0$

**Output:** samples  $x_{1:M}$

```

1: for  $i = 1, \dots, M$  do
2:   sample  $\xi \sim \mathcal{N}(0, 2(1 - \hat{\beta})\eta)$ 
3:    $x_i \leftarrow x_i - \eta \nabla \hat{U}(x_i) + \xi$ 
4: end for
    
```

---

still tends to behave similarly to a random walk once it has converged, resulting in high autocorrelation of samples. We can reintroduce momentum with second order Langevin simulations to avoid this behaviour.

Euler–Maruyama discretization for second order Langevin dynamics gives the update rule

$$\xi_t \sim \mathcal{N}(0, 2\gamma\epsilon), \quad (2.49)$$

$$v_t = v_{t-1} - \epsilon \gamma v_{t-1} - \epsilon \nabla U(x_{t-1}) + \xi_t, \quad (2.50)$$

$$x_t = x_{t-1} + \epsilon v_t. \quad (2.51)$$

The resulting algorithm is called stochastic gradient Hamiltonian Monte Carlo (SGHMC) and is summarized in Algorithm 4. SGHMC was introduced by [Chen \*et al.\* \(2014\)](#). To guarantee convergence we can again use the Robbins–Monro step-size conditions for  $\epsilon$ . Since  $v$  is an auxiliary variable and is not needed for MH corrections as it is in HMC, it is somewhat neater to relabel  $v \leftarrow \epsilon v$ , and redefine the learning rate  $\eta = \epsilon^2$  and momentum decay  $\alpha = \gamma \epsilon$ .

$$\xi_t \sim \mathcal{N}(0, 2\alpha\eta), \quad (2.52)$$

$$v_t = (1 - \alpha)v_{t-1} - \eta \nabla U(x_{t-1}) + \xi_t, \quad (2.53)$$

$$x_t = x_{t-1} + v_t. \quad (2.54)$$

The above update rule is exactly the update rule for (stochastic) gradient descent with momentum ([Sutskever \*et al.\*, 2013](#); [Chen \*et al.\*, 2014](#)), except for the injection noise  $\xi_t$ . One benefit of this is that SGLD and SGHMC can be used with existing stochastic gradient descent implementations just by adding  $x_t \cdot \xi_t$  to the loss function (potential energy). Setting  $\alpha$  to one gives the same update equation as SGLD, i.e. the infinite friction limit.

Just as with SGLD, the parameter  $\hat{\beta}$  in Algorithm 4 controls the stochastic gradient variance. It is meant to be an estimate of the variance of the stochastic

gradients, which may be calculated adaptively or user-specified. We note that here  $\hat{\beta}$  is used slightly differently than in the original version of SGHMC. In our case,  $\hat{\beta}$  is meant to be an estimate of  $\text{Var}[\nabla U]$ , while [Chen \*et al.\* \(2014\)](#) introduce it such that  $\hat{\beta}\eta$  is an estimate of  $\text{Var}[\eta\nabla U]$ . Similarly to HMC,

---

**Algorithm 4** Stochastic Gradient Hamiltonian Monte Carlo
 

---

**Input:** unbiased potential energy estimate  $\hat{U}(x)$ , learning rate  $\eta$ , momentum decay  $\alpha$ , noise estimate  $\hat{\beta}$ , initial state  $x_0$

**Output:** samples  $x_{1:M}$

```

1: sample  $v \sim \mathcal{N}(0, \eta)$ 
2: for  $i = 1, \dots, M$  do
3:   optionally resample momenta  $v \sim \mathcal{N}(0, \eta)$ 
4:   sample  $\xi \sim \mathcal{N}(0, 2(\alpha - \hat{\beta}\eta)\eta)$ 
5:    $v \leftarrow v - \alpha v - \eta\nabla\hat{U}(x_i) + \xi$ 
6:    $x_i \leftarrow x_i + v$ 
7: end for
    
```

---

various extensions to SGHMC exist, such as Riemannian manifold SGHMC which adaptively estimates the local potential energy curvature and stochastic gradient Nosé–Hoover dynamics which couples the system to an external heat source to improve convergence ([Ma \*et al.\*, 2015](#)).

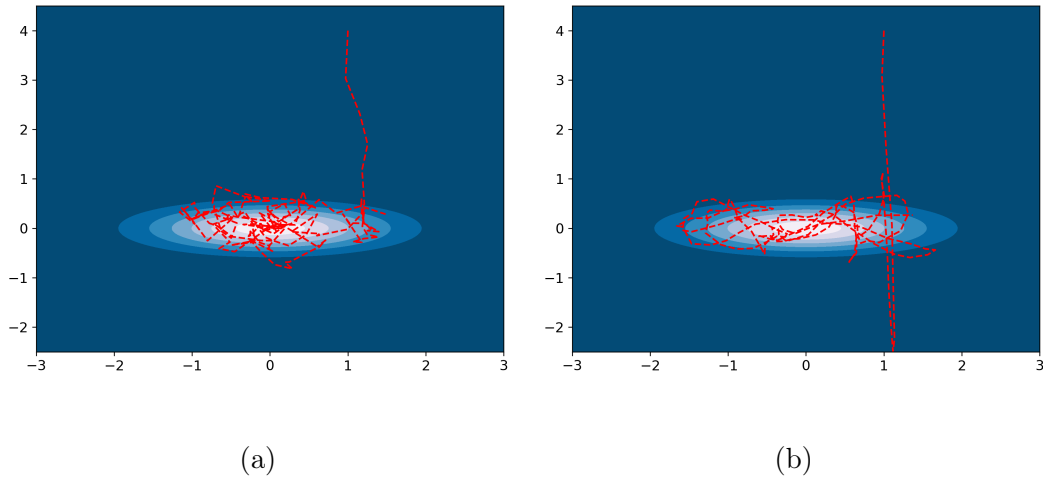


Figure 2.2: (a) shows a trajectory generated by stochastic gradient Langevin dynamics (Algorithm 3) and (b) shows a trajectory generated by stochastic gradient Hamiltonian Monte Carlo (Algorithm 4).

Figure 2.2 shows a comparison of trajectories generated with SGLD and SGHMC. Since SGLD uses gradients, it quickly descends to the minimum of the potential energy, after which it diffuses around exhibiting behaviour similar to a random walk. SGHMC on the other hand uses momentum, and so it has smoother trajectories and travels further distances, covering larger areas of the distribution. See [Chen \*et al.\* \(2014\)](#); [Springenberg \*et al.\* \(2016\)](#); [Ma \*et al.\* \(2015\)](#) for a more in-depth analysis of SGHMC and its parameters.

## Chapter 3

# Algorithms For Evidence Estimation

This chapter will discuss several Monte Carlo algorithms for evidence estimation. The algorithms presented are able to produce accurate estimates for a large variety of models in many contexts. Each algorithm has its own advantages and domain of specialized problems for which it is optimized. Both nested sampling and annealed importance sampling are suited to partition function estimation in statistical physics as they make no assumption about the structure of the likelihood function and so can be equally well applied to integrating the Boltzmann factor.

Nested sampling is incredibly robust to phase changes<sup>1</sup> and pathological likelihood functions but is typically quite difficult to implement. Annealed importance sampling on the other hand is much easier to implement in its basic form but requires careful tuning of the algorithm parameters in order to produce reliable and accurate estimates.

Sequential Monte Carlo is not really an algorithm in itself but rather a general framework for Monte Carlo estimation much like Markov chain Monte Carlo. Most sequential Monte Carlo algorithms are applied to online filtering problems, especially for Markov latent variable models. For the most part, our interest is in estimating evidence for parametric models without latent variable so we will only discuss the basics of sequential Monte Carlo in a limited context, focusing on the most relevant aspects.

---

<sup>1</sup>Non-analytic behaviour at some temperature.

## 3.1 What Makes Evidence Estimation Difficult?

Before considering the more successful algorithms for evidence estimation, we briefly outline the structural difficulties of doing so and show by example how two simple approaches fail miserably to provide reliable evidence estimates.

### 3.1.1 Evidence and Entropy

Evidence is the integral of the likelihood with respect to the prior

$$\mathcal{Z} := p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta) d\theta. \quad (3.1)$$

It is the probability that a model assigns to the sequence of values  $\mathcal{D}$  which were observed.

Now consider the true data generating distribution; the data set is generated according to this distribution through some physical process. This distribution is unobservable and we generally propose some model which we hope is flexible enough that, in the large data limit, its posterior predictive distribution matches closely with the true data generating distribution for each new observation. We will assume that the data is independently and identically distributed. If this is not the case, we can shuffle and resample the data so that it is, otherwise the following argument can be generalized in certain cases. For any reasonably sized data set, the sequence of observations will, with overwhelming probability, lie in the typical set (MacKay, 2002, chapter 4). The typical set consists of those outcome sequences where the relative frequency of any particular outcome in the sequence is approximately equal to the probability of that outcome under the generating distribution. For example, for binary outcomes, if the generating process has a probability of 0.25 of producing a zero, and a probability of 0.75 of producing a one, then the typical set is the collection of sequences for which approximately one quarter of the elements are zero and three quarters are one. Each sequence in the typical set has probability (density; in the case of continuous data) on the order of

$$p_{\text{true}}(\mathcal{D}) \approx e^{-NH}, \quad (3.2)$$

where  $H$  is the entropy of the data generating distribution if the data is discrete and  $H$  is the differential entropy with respect to the Lebesgue measure in the



case of continuous data:<sup>2</sup>

$$H = - \sum_y p(y) \log p(y) \quad \text{or} \quad H = - \int p(y) \log p(y) dy. \quad (3.3)$$

The probability of observing any particular sequence of outcomes, decreases exponentially with the number of observations, and since the true data generating distribution is a better description of the data than any model we might come up with, we expect that the evidences of our models would be upper bounded by this extremely tiny probability. Probabilities are non-negative, so for any model, we expect the evidence to obey the following inequality,

$$0 \leq \mathcal{Z} \leq e^{-NH}. \quad (3.4)$$

If the uncertainty of the evidence estimator is larger than or comparable to the estimator itself, then it is practically useless. Estimating extremely small, but positive quantities poses a challenge for Monte Carlo integration. This same problem arises in rare event simulations (C  rou and Guyader, 2007), where one typically wants to estimate extremely tiny probabilities of events lying in the tails of distributions.

### 3.1.2 Likelihood-Weighted Prior Sampling

For certain simple models, the evidence can be calculated analytically, but for almost all interesting or realistic models, the evidence is analytically intractable. In this case one has to resort to numerical methods. In one or two dimensions quadrature can be used, but the time complexity of quadrature algorithms increases exponentially with the number of dimensions, rendering them infeasible for large problems. Instead, for high dimensional integration, one typically resorts to Monte Carlo methods. To use Monte Carlo integration, the evidence is expressed as an expectation, for example

$$\mathcal{Z} = \mathbb{E}_{p(\theta)} [p(\mathcal{D}|\theta)], \quad (3.5)$$

---

<sup>2</sup>The entropy is taken with respect to the Lebesgue measure in the continuous case because the probability density  $p_{\text{true}}(\mathcal{D})$  is defined in terms of the product Lebesgue measure. We could replace this by another measure in both the entropy definition and the probability density definition, provided we use the same measure.

and an estimator is found which is consistent and preferably unbiased. The simplest example of an evidence estimator is likelihood weighted prior sampling

$$\hat{Z} := \frac{1}{M} \sum_{i=1}^M p(\mathcal{D}|\theta_i), \quad \theta_i \stackrel{\text{i.i.d.}}{\sim} p(\theta). \quad (3.6)$$

Despite being unbiased, the value of  $M$  required for the estimator to converge on realistic models is typically so large that the Monte Carlo simulation would not complete in any reasonable amount of time.

In typical Bayesian inference problems, the prior is quite diffuse to allow the model to learn from the data without introducing unjustified biases, while the likelihood is usually very peaked: usually only a very small volume of the prior is covered by any significant likelihood values. This means that a sample from the prior will typically not coincide with a region of significant likelihood and results in a distribution for the estimator  $\hat{Z}$  which has a very long tail, with the bulk of the distribution lying far below the actual evidence value. The long tail of this distribution is the reason for the estimator still being unbiased despite the high probability to greatly underestimate.

As an example, Figure 3.1 shows the likelihood and prior for a simple Gaussian–Gaussian model with 100 data points. The data was generated by sampling

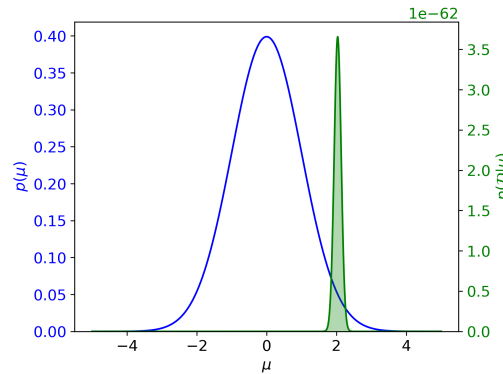


Figure 3.1: The prior and likelihood for a Gaussian–Gaussian model with 100 data points. The prior volume over the shaded region is the evidence.

a Gaussian with mean  $\mu_{\text{true}}=2$  and unit variance. The model applied to this data can be summarised as

$$p(\mu) = \mathcal{N}(\mu|0, 1), \quad p(y_n|\mu) = \mathcal{N}(y_n|\mu, 1) \quad \text{i.i.d. for all } n. \quad (3.7)$$

Figure 3.2a shows the corresponding histogram of the estimator in Equation 3.6. The exact evidence for this model is of the order of  $10^{-64}$ , an ex-

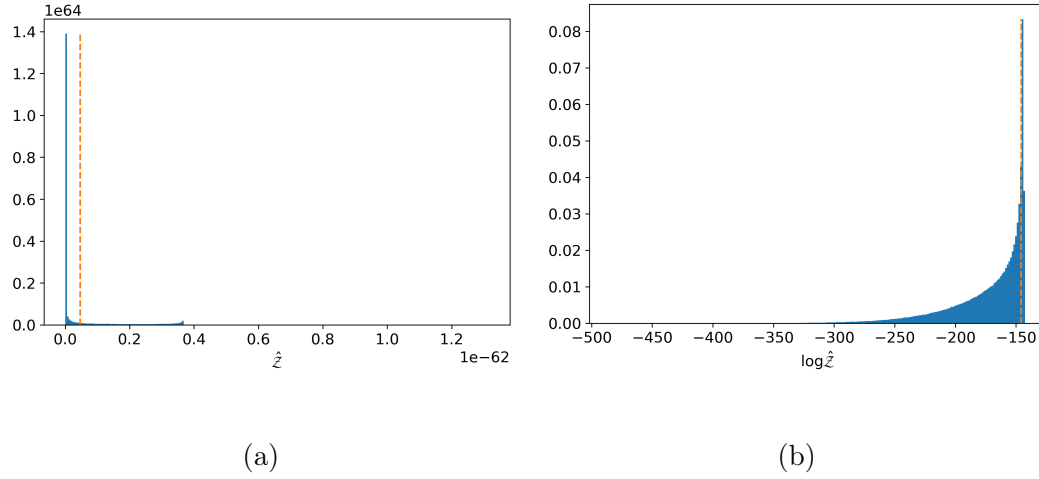


Figure 3.2: (a) shows a normalized histogram with 1000 bins of the evidence estimated by directly averaging the likelihood over  $M = 10$  independent prior samples. (b) shows the same histogram for the log of the estimator. The dashed orange lines are the exact evidence.

tremely small number,<sup>3</sup> but the estimators tend to be so much smaller than even this value, that they default numerically to an exact zero as illustrated in Figure 3.2a.

Evidences and their estimators tend to be extremely small numbers. Since this is the case we will usually be interested in log-evidence or log-evidence per data point rather than the evidence itself. This, however, introduces a new bias, since unbiased evidence-estimators result in biased log-evidence estimators. Bias may not matter if the estimator is consistent and we are only interested in evidence ratios (Bayes Factors) rather than absolute values. However, for use in weighted model combination, we would typically prefer unbiased evidence estimators, not unbiased estimators of the logarithm.

The histogram in Figure 3.2a is for the simple estimator in Equation 3.6 using only  $M = 10$  samples. Naturally we expect that the accuracy will improve if we increase the number of samples; however, the distribution of this estimator

<sup>3</sup>To put that into perspective, the radius of a proton measured in parsecs is  $2.84 \times 10^{-32}$  and the mass of an electron measured in solar masses is  $4.58 \times 10^{-61}$  (according to Wolfram Alpha).

is so skewed that the central limit theorem approximation will be poor until  $M$  is extremely large. The above example is possibly one of the simplest models of all, and 100 data points is not a large number by modern standards, so naturally if simple prior sampling with Equation 3.6 is inaccurate for this model, we cannot expect it to work well on complex models with many parameters and for many data points.

### 3.1.3 Harmonic Mean

No discussion of evidence estimators would be complete without mentioning the harmonic mean estimator. The idea behind estimating evidence with harmonic averaging is based on the following equation

$$\mathbb{E}_{p(\theta|\mathcal{D})} \left[ \frac{1}{p(\mathcal{D}|\theta)} \right] = \int \frac{1}{p(\mathcal{D}|\theta)} \frac{p(\mathcal{D}|\theta)p(\theta)}{\mathcal{Z}} d\theta = \frac{1}{\mathcal{Z}}. \quad (3.8)$$

This equation only applies when the posterior has the same support as the prior and therefore requires that the likelihood be non-zero everywhere. This equation suggests the following estimator

$$\hat{\mathcal{Z}}_{\text{HME}} := \left( \frac{1}{M} \sum_{i=1}^M \frac{1}{p(\mathcal{D}|\theta_i)} \right)^{-1}, \quad (3.9)$$

where  $\theta_i$  are drawn from the posterior distribution, probably using some MCMC method. This estimator is biased; its reciprocal is an unbiased estimator for the reciprocal of the evidence.

In practice it tends to be very inaccurate and often has infinite variance. For this reason it has been criticized as the “worst Monte Carlo algorithm ever” (Neal, 2008). The main flaw of the harmonic mean estimator can be put down to the fact that the samples are drawn from the posterior, which is insensitive to the prior and therefore also the evidence.

A histogram of the evidence estimates produced by the harmonic mean by exactly sampling the posterior of the Gaussian–Gaussian model can be seen in Figure 3.3a. For this simple example the harmonic mean typically overestimates the evidence by at least two orders of magnitude. This histogram was generated by exactly sampling the posterior; however, for realistic models this is usually not possible and instead MCMC algorithms are used to generate approximate samples. Since this results in samples which are approximate

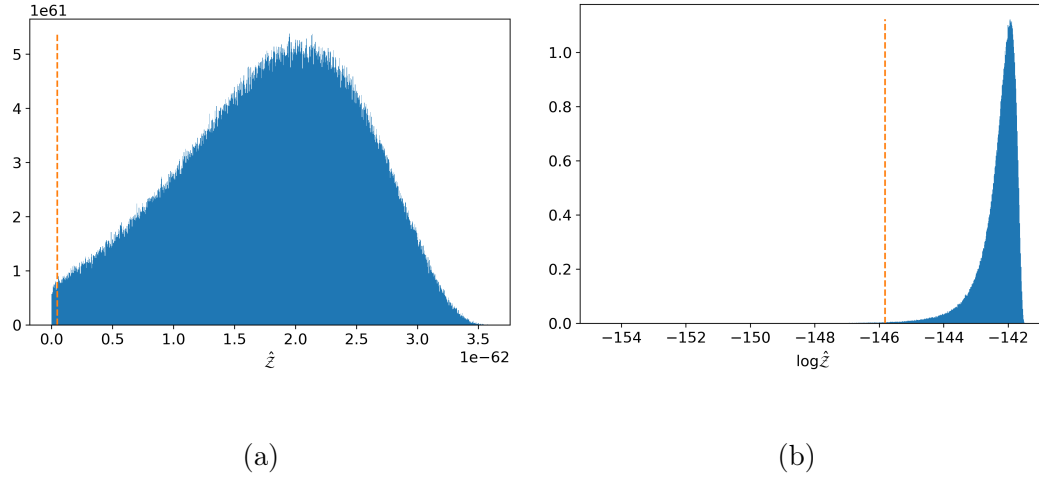


Figure 3.3: (a) shows a normalized histogram with 1000 bins of the harmonic mean estimator for the Gaussian–Gaussian model using  $M = 10$  posterior samples. (b) shows the same histogram for the log of the harmonic mean estimator. The dashed orange lines are the exact evidence.

and correlated, one would expect that the accuracy of the harmonic mean estimator will be made worse by doing this. MCMC algorithms tend to have difficulty moving between modes; therefore if MCMC is used to approximately sample from a multi-modal posterior then the distribution of the harmonic mean estimator can change drastically. This tends to be less of a problem for other evidence estimators but can be fatal for the harmonic mean.

### 3.1.4 Sandwiching and Uncertainty Estimation

Evidence estimators more generally tend to consistently underestimate or consistently overestimate (Grosse *et al.*, 2015). Evidence estimators are positive real random variables and so we can apply Markov’s inequality

$$\Pr \left[ \hat{\mathcal{Z}} \geq \mathbb{E} \left[ \hat{\mathcal{Z}} \right] e^{\alpha} \right] = \Pr \left[ \log \hat{\mathcal{Z}} \geq \log \mathbb{E} \left[ \hat{\mathcal{Z}} \right] + \alpha \right] \leq e^{-\alpha}. \quad (3.10)$$

If the estimator is unbiased then we can be certain that it will not overestimate the exact evidence by more than a few nats. The same inequality can be applied to the reciprocal of the harmonic mean estimator to show that it will almost never underestimate the exact evidence by more than a few nats (assuming exact posterior sampling is used). For unbiased estimators, the bulk of the distribution lying far below the exact evidence means that the estimator must have a large variance. Furthermore, since the distributions of

unbiased evidence estimators tend to be skewed in this way, estimating the evidence many times and then calculating the sample variance tends to give no indication of the accuracy of the estimator and only an indication of the typical values that the estimator produces which may be very different from the exact evidence.

[Grosse \*et al.\* \(2015\)](#) propose an approach called bidirectional Monte Carlo, which produces two evidence estimators which can be based on either annealed importance sampling or sequential Monte Carlo. The first is unbiased and so is very unlikely to overestimate. They call this estimator a stochastic lower bound. The second estimator is biased and based on reversing annealed importance sampling or a harmonic mean variant of sequential Monte Carlo. The second estimator is very unlikely to underestimate, and so they call it a stochastic upper bound. The difference between these two estimators gives a fair and prudent approximation of their accuracy.

Unfortunately the stochastic upper bound requires an exact posterior sample in order to theoretically guarantee a low probability of underestimating. This means that it cannot be used to quantify accuracy on most realistic models with real data. However it can be used on simulated data sets. If one samples the parameters from the prior, and then samples data points using these parameters, the initially sampled parameters will be an exact posterior sample given the generated data.

[Grosse \*et al.\* \(2015\)](#) recommend this as a method of testing the typical accuracy of other evidence estimators by comparing their estimates to the bidirectional Monte Carlo bounds on artificially generated data for models of interest. Unfortunately, if the generated data is radically different from data being modeled, then the accuracy of the estimator of interest may be much lower on the real data than on the simulated data. This approach has been successfully applied to variational autoencoders ([Cremer \*et al.\*, 2018](#)).

## 3.2 Nested Sampling

Nested sampling (NS) ([Skilling, 2004, 2006](#)) is a general purpose Monte Carlo algorithm for estimating integrals. Nested sampling was introduced to estimate Bayesian evidence and is therefore particularly well-suited to estimating

expectations of extremely peaked functions. In the same way that quadrature can be thought of as a direct numeric approximation to Riemann integration, NS can be thought of as directly approximating Lebesgue integration. This is done by selecting successive subsets  $S_k$  of the integration domain for  $k = 1, \dots$  and approximating the integrand, i.e. the likelihood, by a sum of piecewise constant function on these sets.

Consider a function of the form

$$l(\theta) = \sum_k \alpha_k \mathbf{1}_{S_k}(\theta), \quad (3.11)$$

where

$$\mathbf{1}_A(\theta) := \begin{cases} 1 & \text{if } \theta \in A \\ 0 & \text{otherwise} \end{cases} \quad (3.12)$$

is the indicator function on the set  $A$  and the sum is finite. In measure theory, this type of function is called a simple function and its integral with respect to an arbitrary measure<sup>4</sup>  $p$  is defined by linearity

$$\int l(\theta) p(\theta) d\theta := \sum_k \alpha_k p(S_k). \quad (3.13)$$

The Lebesgue integral of an arbitrary non-negative real-valued measurable<sup>5</sup> function  $f$  is defined to be the supremum of integrals of non-negative simple functions which are bounded above by  $f$ .

$$\int f(\theta) p(\theta) d\theta := \sup \left\{ \int l(\theta) p(\theta) d\theta \mid 0 \leq l \leq f, l \text{ simple} \right\}. \quad (3.14)$$

Therefore we may approximate the integral of a function, such as the likelihood of a Bayesian model, by the integral of a carefully chosen simple function. This definition can further be extended to functions which may be negative or vector valued as well as to signed or complex measures through linearity requirements. See [Rudin \(1987\)](#) for a formal treatment of integration.

We noted in [Section 3.1](#) that the main source of inaccuracy in directly averaging likelihood values over prior samples is due to the fact that the regions of high likelihood, which contribute most to the integral, have a very low probability of being sampled. To combat this problem, NS chooses the sets for the simple

<sup>4</sup>A countably additive non-negative function of sets.

<sup>5</sup>Well behaved in the sense that we can formulate a consistent definition of its integral.

function approximation to be nested,  $S_1 \supset S_2 \supset S_3 \supset \dots$  where the values of the likelihood on the set  $S_k$  are greater than the values of the likelihood on the set  $S_m \setminus S_k$  when  $k > m$ . These sets are generated by independently sampling  $\theta_k$  from the prior under the constraint that each new sample have higher likelihood than any previous samples  $\theta_j, j < k$ . In other words, the  $k^{\text{th}}$  sample point  $\theta_k$  is generated from the prior restricted to  $S_{k-1}$  and the set  $S_k$  is then defined to be<sup>6</sup>

$$S_k := \{\theta \mid p(\mathcal{D}|\theta) > p(\mathcal{D}|\theta_k)\}. \quad (3.15)$$

This process is repeated until a likelihood value is reached which is high enough for the integral approximation to be accurate. Since the sets  $S_k$  are nested, the  $k^{\text{th}}$  coefficient of the simple function approximation is the difference between the likelihood of the  $k^{\text{th}}$  sample and the previous one:

$$l(\theta) = \sum_k \Delta\lambda_k \mathbf{1}_{S_k}(\theta), \quad \text{where} \quad \lambda_k := p(\mathcal{D}|\theta_k), \quad (3.16)$$

and  $\Delta\lambda_k = \lambda_k - \lambda_{k-1}$ . The integral of  $l(\theta)$  over the prior will be a lower bound on the evidence. In the continuous limit, the sum  $\sum_k \Delta\lambda_k p(S_k)$  will converge to the evidence.

Unfortunately, integrating  $l(\theta)$  with Equation 3.13 is not so simple since we do not know the prior mass of the nested sets  $p(S_k)$ . Fortunately, we do know the probability distribution of  $p(S_k)$  when  $\theta_k$  is sampled from the prior restricted to  $S_{k-1}$ . Since the sets  $S_k$  are defined to be nested,  $p(S_k)$  is a kind of cumulative distribution function in  $\theta_k$  and so its distribution when conditioned on  $S_{k-1}$  is uniform. To see this, define  $X_k := p(S_k)$  and  $\rho_k := p(\theta \in S_k | \theta \in S_{k-1}) = X_k/X_{k-1}$ . The set  $S_k$  is the preimage under the likelihood function of the interval  $(\lambda_k, \infty)$ , therefore the probability  $\rho_k$  that a  $\theta$  sampled from the prior restricted to  $S_{k-1}$  is in the set  $S_k$  is just the probability that its corresponding likelihood value is larger than  $\lambda_k$ . Then

$$\rho_k = \int_{\lambda_k}^{\infty} p_{\lambda}(\lambda') d\lambda', \quad (3.17)$$

---

<sup>6</sup>For discrete likelihoods, the  $>$  may need to be replaced by  $\geq$ . Discrete likelihoods require a more intricate analysis in the case that the newly sampled  $\theta_k$  have the same likelihood as  $\theta_{k-1}$ . Here we assume that the likelihood is continuous and that the probability of sampling a point with equal likelihood is zero.



where  $p_\lambda$  is the (unknown) distribution of the likelihood value of a point  $\theta$  sampled from the prior restricted to  $S_{k-1}$ .  $\rho_k$  is a monotonic decreasing function of  $\lambda_k$ , and so we can use the variable transformation rule to calculate its probability density

$$p(\rho) = p_\lambda(\lambda) \left| \frac{d\rho}{d\lambda} \right|^{-1}, \quad (3.18)$$

$$= p_\lambda(\lambda) | -p_\lambda(\lambda) |^{-1}, \quad (3.19)$$

$$= 1. \quad (3.20)$$

Therefore we have the conditional distribution for  $X_k$

$$X_k \sim U(0, X_{k-1}). \quad (3.21)$$

Knowing this probability distribution allows us to approximate the integral with the following estimator:

$$\hat{\mathcal{Z}} := \sum_k \Delta \lambda_k \mathbb{E}[X_k], \quad (3.22)$$

where the expectation is taken over the joint distribution of all the samples  $\theta_k$ .

We can improve the tightness of the integral approximation and simultaneously reduce the variance of the evidence estimator by using multiple constrained prior samples (live particles) at each step instead of only one. In this case we may choose  $\lambda_k$  to be the smallest of all the likelihood values and  $\theta_k$  to be the corresponding particle. The next iteration, we may keep all the other sampled particles, since they are already within the set  $S_k$  and only generate a single new sample. The first result of using multiple live particles is that each set  $S_k$  typically becomes larger and so in the limit of an infinite number of live particles, the sum in Equation 3.22 converges exactly to the evidence. The second result of taking the smallest likelihood is that  $\rho_k$  becomes beta distributed (the maximum order statistic of the uniform distribution) (Linden *et al.*, 2014, chapter 7). For  $M$  particles

$$\frac{X_k}{X_{k-1}} \sim \beta(M, 1). \quad (3.23)$$

As  $M$  approaches infinity, the variance of the estimator approaches zero. The expectation now becomes

$$\mathbb{E}[X_k] = \frac{M}{M+1} \mathbb{E}[X_{k-1}] = \left( \frac{M}{M+1} \right)^k \quad (3.24)$$

Instead of approximating  $\rho_k$  as  $M/(M+1)$ , Skilling (2006) uses  $e^{-1/M}$ . The difference of these values is

$$\frac{M}{M+1} - e^{-1/M} = \frac{1}{2M^2} + \mathcal{O}\left(\frac{1}{M^3}\right), \quad (3.25)$$

which is exponentially suppressed in Equation 3.24, even for small values of  $M$ . The error in this approximation, as well as the error from approximating the likelihood as a step function tend to be dominated by the variance of the estimator, which goes to zero as  $M$  increases (Linden *et al.*, 2014, chapter 31). We use the  $e^{-1/M}$  factor in our code.

An alternative evidence estimator based on the nested sampling process is

$$\hat{\mathcal{Z}} := \sum_k \lambda_k w_k \quad \text{where} \quad w_k = \mathbb{E}[X_k - X_{k+1}]. \quad (3.26)$$

This equation has a similar form to an importance weighted estimator, and it is more natural when one also wants to estimate posterior expectations. This alternate estimator is equal to Equation 3.22 up to a term  $\lambda_K \mathbb{E}[X_K]$  where  $K$  is the index of the last iteration. This term is negligible since  $X_k$  decreases exponentially with  $k$ . We can similarly get an unbiased estimator for a tight upper bound by replacing  $w$  in the above equation with  $\mathbb{E}[X_{k-1} - X_k]$ . In the limit  $M \rightarrow \infty$ , the upper bound estimator also converges to the true evidence.

The samples generated as a byproduct of using NS to estimate evidence can be used to estimate posterior expectations. The samples are not distributed, even approximately, according to the posterior since as the number of iterations increases, all subsequent samples cluster at the peak of the likelihood function. Posterior expectations are therefore approximated not by averaging over the samples equally but by using a weighted average, where the weight is given by the (approximate) posterior mass in the shell  $S_k \setminus S_{k-1}$ . For a test function  $h$ , the posterior expectation can be estimated by

$$\hat{H} := \frac{1}{\hat{\mathcal{Z}}} \sum_k h(\theta_k) \lambda_k w_k. \quad (3.27)$$

In this case, using the evidence estimator in Equation 3.26 is desirable, since it provides the correct normalization in the above equation, i.e. for  $h(\theta) = 1$ , the above equation correctly yields  $\hat{H} = 1$ , but using the estimator in Equation 3.22 would not.

Nested sampling is summarized using Equation 3.26 in Algorithm 5; where we have used the relationship  $w_k = w_{k-1}M/(M+1)$  and  $X_0 = 1$ .

---

**Algorithm 5** Nested Sampling

---

**Input:** Data  $\mathcal{D}$ , number of particles  $M$ , pdfs  $p(y|\theta), p(\theta)$

**Output:**  $\hat{\mathcal{Z}}$  evidence estimator

```

1:  $\hat{\mathcal{Z}} \leftarrow 0$ 
2:  $w \leftarrow 1 - \frac{M}{M+1}$ 
3:  $\forall_i$ : sample  $\theta_i \sim p(\theta)$ 
4: repeat
5:    $j \leftarrow \operatorname{argmin}_i \{p(\mathcal{D}|\theta_i)\}$ 
6:    $\lambda \leftarrow p(\mathcal{D}|\theta_j)$ 
7:    $\hat{\mathcal{Z}} \leftarrow \hat{\mathcal{Z}} + \lambda w$ 
8:   replace particle  $\theta_j \sim p(\theta)$  subject to  $p(\mathcal{D}|\theta) > \lambda$ 
9:    $w \leftarrow w \frac{M}{M+1}$ 
10: until converged
11: return  $\hat{\mathcal{Z}}$ 

```

---

There is still one important detail which needs to be discussed. NS requires us to sample *independently* from the prior under the constraint of increasing likelihood. Sampling under constraints is, in general, a difficult problem, even for simple constraints such as  $\theta > 0$ . The constraint of increasing likelihood results in sets  $S_k$  which may have very complex shapes, and so the only general way we can reliably guarantee that samples are generated independently according to the constrained prior is to repeatedly generate independent samples from the unconstrained prior until one is generated which has sufficient likelihood. Although this rejection sampling approach is valid, it is incredibly inefficient and quickly becomes intractable for problems in more than one or two dimensions. Since the likelihood is extremely peaked, the number of trials required to accept even a single sample can be in the trillions.

In practice, Markov kernels are used to generate new samples, starting each Markov transition from a random live particle. It is possible to enforce the likelihood constraints for MH based algorithms by simply rejecting steps that cross the likelihood boundary. Since the initial particles are drawn exactly from the prior, applying a Markov kernel which has the constrained prior as its stationary distribution results in a new sample which is also distributed

according to the constrained prior.<sup>7</sup> Momentum based kernel such as HMC can be used by reflecting off likelihood constraint boundaries (Skilling, 2012). For momenta  $v$ , this can be done by updating the momenta as follows

$$v \leftarrow v - 2\hat{n}(v \cdot \hat{n}), \quad (3.28)$$

where  $\hat{n}$  is the unit vector parallel to the gradient of the likelihood  $\nabla_{\theta} p(\mathcal{D}|\theta)$ .

The derivation of the distribution of  $X$  is based on the assumption that the samples are generated independently. If the Markov chain is allowed to burn in for long enough to greatly reduce the correlation then the approximation will still be fairly good. However, one still needs to be careful since it is unlikely that the particles will be able to cross from one mode of the distribution to another, and the autocorrelation of the chain may be increased by repeatedly rejecting steps that cross the likelihood constraint boundary. The unfortunate consequence is that, except for in a few specialized situations, any practical implementation of NS loses its theoretical guarantees.

### 3.3 Annealed Importance Sampling

We noted before that directly averaging the likelihood over prior samples yields a high variance estimator which is completely unreliable. There are many methods to reduce the variance of Monte Carlo estimators, one of which is importance sampling. Importance sampling is a simple technique which can easily be used to augment sampling algorithms. The basic principle of importance sampling relies on the equality

$$\mathbb{E}_{p(\theta)} [h(\theta)] = \mathbb{E}_{q(\theta)} \left[ h(\theta) \frac{p(\theta)}{q(\theta)} \right] \quad (3.29)$$

for any distribution  $q$  whose support covers the support of  $p$ . We can therefore replace any unbiased estimator for  $H := \mathbb{E}_{p(\theta)} [h(\theta)]$ , where  $h$  and  $p$  are allowed to be arbitrary, with another estimator simply by replacing  $p(\theta)$  with  $q(\theta)$  and  $h(\theta)$  with  $h(\theta)p(\theta)/q(\theta)$ . The resulting estimator will be unbiased but it will generally have lower or higher variance than the original estimator. This idea is very general and can be applied to MCMC sampling, providing the same guarantees as direct MCMC, or any other unbiased estimator. While there

---

<sup>7</sup>Assuming the proposal distribution has unbounded support.

exists a host of research on importance sampling techniques, much of it is beyond the scope of this thesis. Corresponding to the simple estimator of  $\mathbb{E}_{p(\theta)}[h(\theta)]$  based on direct sampling

$$\hat{H}_{\text{DS}} := \frac{1}{M} \sum_{i=1}^M h(\theta_i), \quad \theta_i \stackrel{\text{i.i.d.}}{\sim} p(\theta), \quad (3.30)$$

given a proposal distribution  $q$ , also called the importance distribution, we obtain the simple importance sampling estimator

$$\hat{H}_{\text{IS}} := \frac{1}{M} \sum_{i=1}^M h(\theta_i) w_i, \quad \theta_i \stackrel{\text{i.i.d.}}{\sim} q(\theta), \quad (3.31)$$

where  $w_i = p(\theta_i)/q(\theta_i)$  are the importance weights, whose expectation with respect to  $q$  is one. The difference in the variance of the direct sampling estimator and the importance sampling estimator is

$$\frac{1}{M} \mathbb{E}_{p(\theta)} \left[ h^2(\theta) \left( 1 - \frac{p(\theta)}{q(\theta)} \right) \right]. \quad (3.32)$$

This quantity is positive if the importance sampler has a lower variance than the original estimator, and it is negative if the importance sampler has a higher variance. If  $q(\theta)$  is small in regions where  $p$  places considerable mass, then the ratio  $p(\theta)/q(\theta)$  will contribute significantly to the above expectation resulting in a higher variance for the importance sampling estimator than the original. However if  $q$  places considerable mass in regions where  $p(\theta)$  is small, then the variance of the importance sampling estimator will not be penalized as heavily. For this reason, it is usually recommended that  $q$  be a distribution with heavier tails than  $p$ . Naturally, the variance of the estimator will depend strongly on the function  $h$ . Proposal distributions which place considerable mass in regions which contribute most to the expectation have lower variance than those that do not.

Theoretically, the minimum variance importance sampling estimator samples from the proposal distribution would be

$$q_{\text{opt}}(\theta) = \frac{h(\theta)p(\theta)}{\mathbb{E}_{p(\theta)}[h(\theta)]}. \quad (3.33)$$

The corresponding simple importance sampling estimator is then

$$\hat{H}_{\text{opt}} := \frac{1}{M} \sum_{i=1}^M \mathbb{E}_{p(\theta)}[h(\theta)], \quad \theta_i \stackrel{\text{i.i.d.}}{\sim} q_{\text{opt}}(\theta). \quad (3.34)$$

This estimator simply adds a number of constants and has zero variance. It also requires knowledge of the exact value of the expectation which we are trying to estimate in the first place and is therefore not very practical. In practice, for simple importance sampling, we also need to be able to generate samples from the proposal distribution  $q$  and so this limits our choice to simple common distributions. The proposal distribution needs to overlap well with  $h(\theta)p(\theta)$  to be effective, but in high dimensions with a limited choice of common distributions, this tends to be difficult or impossible to achieve.

Importance sampling can also be iterated with a sequence of importance distributions. For two distributions  $q$  and  $r$  such that the support of  $q$  covers the support of  $p$  and the support of  $r$  covers the support of  $q$ ,

$$\mathbb{E}_{p(\theta)} [h(\theta)] = \mathbb{E}_{q(\theta)} \left[ h(\theta) \frac{p(\theta)}{q(\theta)} \right] = \mathbb{E}_{r(\theta)} \left[ h(\theta) \frac{p(\theta)}{q(\theta)} \frac{q(\theta)}{r(\theta)} \right]. \quad (3.35)$$

This provides no benefit for the simple importance sampling estimator as the intermediate distributions in the fractions simply cancel. It can, however, be used in more complex algorithms such as sequential importance sampling and annealed importance sampling (AIS). AIS (Neal, 1998) estimates expectations, by iterated approximate sampling from a sequence of distributions, bridging from a distribution which is easy to sample from to some desired distribution. As before, each distribution in the sequence must have a support which covers the support of the next distribution, and in order to be effective, each distribution in the sequence should be more diffuse than the next but without being too dissimilar.

While the theory of AIS applies to any sequence of intermediate distributions satisfying the support requirement, it is almost always used with a sequence of distributions which are geometrically interpolated between a simple distribution and the desired distribution. AIS does not require the normalization constant of the desired distribution and provides an unbiased estimator of the ratio of the normalizing constants of the desired distribution and the initial importance distribution. Since we want to estimate Bayesian evidence, this will be our main interest in AIS. For Bayesian inference problems, the most natural initial distribution is the prior, the desired final distribution is the posterior, and the sequence of unnormalized geometrically interpolated distributions is

$$f_t(\theta) = p(\mathcal{D}|\theta)^{\lambda_t} p(\theta), \quad (3.36)$$

where  $(\lambda_t)_{t=0}^T$  is any increasing sequence with  $\lambda_0 = 0$  and  $\lambda_T = 1$ . The sequence of  $\lambda$ 's is called the annealing schedule. In the above equation,  $\lambda$  plays a role similar to the inverse temperature in the Boltzmann distribution discussed in Section 2.2; hence the name “annealing”. The annealing schedule controls the number of intermediate distributions as well as the extent to which each consecutive importance distribution differs from the last, and so the accuracy and computational efficiency depend very strongly on the choice of annealing schedule. For now we will assume that the schedule is fixed; later we will discuss how to select the annealing schedule adaptively.

AIS proceeds by initially generating  $M$  particles  $\theta_{1:M}^{(0)}$  — not necessarily independently — from the distribution  $f_0$ , which for our purposes is the prior, and then for each intermediate distribution in the sequence, updating the importance weights and then moving the particles by applying some Markov kernel which leaves the current distribution invariant. The application of a Markov kernel to update the particles allows iterated importance sampling without the computation being redundant as it is in the simple importance sampling estimator. The importance weights are all initialized to 1 and at each time-step  $t$  are updated based on the current particles  $\theta_i^{(t-1)}$  as follows:

$$w_i^{(t)} = w_i^{(t-1)} \frac{f_t(\theta_i^{(t-1)})}{f_{t-1}(\theta_i^{(t-1)})} = w_i^{(t-1)} p(\mathcal{D}|\theta_i^{(t-1)})^{\lambda_t - \lambda_{t-1}}. \quad (3.37)$$

The particles are then updated by sampling  $\theta_i^{(t)}$  from a Markov kernel, based at  $\theta_i^{(t-1)}$ , which has  $f_t$  as its stationary distribution. After each time step  $t$ , the expectation of the importance weights is the normalization constant of  $f_t$ , and so they can be used to estimate the unnormalized expectation of a test function  $h$  over  $f_t$  in a similar manner to Equation 3.31

$$\mathbb{E} \left[ h(\theta_i^{(t-1)}) w_i^{(t)} \right] = \int h(\theta) p(\mathcal{D}|\theta)^{\lambda_t} p(\theta) d\theta. \quad (3.38)$$

Therefore at each time-step, we have the unbiased estimator

$$\hat{H}^{(t)} := \frac{1}{M} \sum_{i=1}^M h(\theta_i^{(t-1)}) w_i^{(t)} \xrightarrow{M \rightarrow \infty} \int h(\theta) p(\mathcal{D}|\theta)^{\lambda_t} p(\theta) d\theta. \quad (3.39)$$

Proof that this estimator is unbiased is based on considering the extended state space of trajectories  $\theta_i^{1:T}$ , and treating the sampling procedure as a simple importance sampler on this space. See Neal (1998) for the detailed proof.

The AIS evidence estimator is the special case  $\hat{\mathcal{Z}} = \hat{H}^{(T)}$  for the test function  $h(\theta) = 1$ . Biased but consistent estimators of posterior expectations of a test function  $h$  can be obtained by simply dividing the unnormalized estimator by the evidence estimator. This can be used for estimating posterior predictive probabilities of a test value  $y'$  by using  $h(\theta) = p(y'|\theta)$ .

At each time step, the Markov transition must exhibit  $f_t$  as its stationary distribution. We can efficiently update the particles using HMC with the following potential energy function

$$U^{(\lambda)}(\theta) = -\lambda \log p(\mathcal{D}|\theta) - \log p(\theta). \quad (3.40)$$

By using HMC instead of some other simple MH algorithm or MALA, the particle is allowed to travel further and more closely converge to  $f_t$ . SGHMC could also be used, however the importance weight updates in Equation 3.37 still require iterating over the whole data set so in this form AIS cannot take full advantage of mini-batching.

By choosing a geometric interpolation of the prior and posterior for the intermediate distributions, we guarantee that the importance distributions will satisfy the support requirement, and we can be fairly certain that each intermediate distribution is more diffuse than the next. Therefore to reduce the variance of the estimators, we should choose the annealing schedule such that each intermediate distribution is as close as possible to the next. A common way to do this is to calculate the empirical effective sample size (ESS) (Kong *et al.*, 1994), and adaptively choose each subsequent  $\lambda$  in the sequence such that the ESS is approximately equal to some user-specified target (Buchholz *et al.*, 2018; Beskos *et al.*, 2016). For importance sampling the ESS is given by

$$\text{ESS} = \frac{(\sum_i w_i)^2}{\sum_i w_i^2}. \quad (3.41)$$

This is a kind of heuristic which can be interpreted loosely as the number of samples from the target distribution which would be required for a naive Monte Carlo estimator to have similar variance to the importance sampled estimator. This interpretation is not always strictly valid, since  $1 \leq \text{ESS} \leq M$  when there are  $M$  particles, and directly sampling the target may have a much larger variance or a much lower variance than an importance sampled estimator. However for AIS it can simply be thought of as the effective number



of particles for calculating expectations with respect to distribution  $f_t$  when the particles are drawn from  $f_{t-1}$ . For the importance weights in AIS, the ESS will depend on the annealing schedule. Defining  $\Delta = \lambda_t - \lambda_{t-1}$ ,

$$\text{ESS}(\Delta) = \frac{(\sum_i \omega_i(\Delta))^2}{\sum_i \omega_i(\Delta)^2}, \quad (3.42)$$

where

$$\omega_i(\Delta) = p(\mathcal{D}|\theta_i^{t-1})^\Delta. \quad (3.43)$$

The annealing schedule can be chosen adaptively by solving for  $\lambda_t$  in the equation  $\text{ESS}(\lambda_t - \lambda_{t-1}) = \text{ESS}^*$  given some target  $\text{ESS}^*$ . ESS is monotonic in  $\Delta$  so a simple bisection search can be used.

---

**Algorithm 6** Annealed Importance Sampling

---

**Input:** Data  $\mathcal{D}$ , number of particles  $M$ , pdfs  $p(\mathcal{D}|\theta), p(\theta)$ , target ESS:  $\text{ESS}^*$

**Output:**  $\hat{\mathcal{Z}}$  evidence estimator

```

1:  $\forall_i$ : sample  $\theta_i \sim p(\theta)$ 
2:  $\forall_i$ :  $w_i \leftarrow 1$ 
3:  $\lambda \leftarrow 0$ 
4: while  $\lambda < 1$  do
5:    $\Delta \leftarrow \text{argmin}_\Delta [\text{ESS}(\Delta) - \text{ESS}^*]$   $\triangleright \Delta \in (0, 1 - \lambda]$ 
6:    $\lambda \leftarrow \lambda + \Delta$ 
7:    $\forall_i$ :  $w_i \leftarrow w_i p(\mathcal{D}|\theta_i)^\Delta$ 
8:    $\forall_i$ :  $\theta_i \leftarrow \text{HMC}(\theta_i, U^{(\lambda)})$   $\triangleright U^{(\lambda)}$  defined in Equation 3.40
9: end while
10: return  $\hat{\mathcal{Z}} = \frac{1}{M} \sum_i w_i$ 

```

---

AIS is summarized in Algorithm 6 where HMC is used to update the particles. One potential benefit of AIS in the context of equilibrium statistical physics simulations is that it provides an estimate of the partition function  $\mathcal{Z}(\lambda)$  and expectations of interest for each value of  $\lambda$  in the annealing schedule, from which interesting properties of the system can be extracted from the intermediate values, such as critical points and phase diagrams. For evidence estimation, on the other hand, this is probably unnecessary since the intermediate distributions have no real meaning to us.

In systems which exhibit phase changes and critical transition, a long burn-in time might be required for particles distributed approximately according to  $f_{t-1}$  to reach  $f_t$ , which may result in high variance of the estimator near to

the critical point. Adaptively annealing helps to reduce these effects but care should still be taken when applying AIS to complex systems. Although these phase changes and critical transition occur frequently in physics models, they are also known to occur in Bayesian models.

### 3.4 Sequential Monte Carlo

The framework of sequential Monte Carlo (SMC) extends the basic ideas from the previous section and as such AIS falls into this class. Many SMC algorithms are conventionally called particle filters and are specifically targeted towards online inference applications. Although the framework of SMC applies equally well to many different model classes, these algorithms are often used in latent variable models with an inherent dynamical structure. Many of these models make a Markov assumption about the latent variables  $\{z_n\}$  and factorize the joint distribution

$$p(\{y_n\}, \{z_n\}, \theta) = p(\theta) \prod_n p(y_n | z_n, \theta) p(z_n | z_{n-1}, \theta), \quad (3.44)$$

The key difference between the model parameters  $\theta$  and the latent variables is that the number of parameters stays constant as the data set size increases, while the number of latent variables grows with the number of data points. The distribution of the latent variables and the parameters can both be inferred through Bayes' theorem, so from a purely probabilistic perspective there is no fundamental difference in the way they are treated. NS and AIS do not assume any specific structure of the likelihood function and so they can trivially be used with these kinds of models by augmenting the parameter space with the latent variables. However, in an online context where new observations are periodically added to the data set, NS and AIS have no way to incorporate the extra dimensionality required to include the new observations and must perform the entire calculation again from scratch.

Particle filtering algorithms are particularly convenient for these kinds of scenarios because they naturally incorporate the dynamics of the model. The general framework of SMC, although specifically aimed at these sorts of problems, works equally well for parametric models which do not have latent variables. In this work we will mainly consider the case where there are no latent

variables thus avoiding many of the intricacies that arise in latent variable models. For an in-depth introduction to SMC, see [Naesseth \*et al.\* \(2019\)](#) and [Doucet \*et al.\* \(2001\)](#).

As in AIS, the key idea behind other SMC algorithms is to use a sequence of intermediate distributions to transition from a simple distribution to a complex target distribution which is of interest. The choice of sequence may be guided by the structure of the problem at hand or chosen simply because it is convenient. For AIS, the sequence of distributions is chosen to be a geometric interpolation of the simple and the desired distribution. In SMC terminology, this is referred to as thermal tempering. For physics simulations, the intermediate distributions have some physical meaning but for Bayesian inference they are not necessarily relevant to the problem.

Another common sequence of distributions is given by Bayesian updating. Assuming conditionally independent data, this can be written

$$f_n(\theta) = p(\theta) \prod_{k \leq n} p(y_k | \theta), \quad (3.45)$$

or

$$f_n(\theta, z_{1:n}) = p(\theta) \prod_{k \leq n} p(y_k | z_k, \theta) p(z_k | z_{k-1}, \theta), \quad (3.46)$$

for Markov latent variable models. Such updating of distributions is particularly meaningful in Bayesian inference problems and enables online estimation. In SMC terminology, this approach is called data tempering. Although similar in nature to thermal tempering, data tempering limits the extent to which the intermittent distributions could be chosen adaptively since the Bayesian updating process is inherently discrete.

Data tempering corresponds to calculating (potentially correlated) importance sampling estimates of predictive distributions  $p(y_n | y_{<n})$  and factoring the evidence as

$$\mathcal{Z} = \prod_n p(y_n | y_{<n}). \quad (3.47)$$

One potential benefit of this formulation is that it reduces the difficult problem of estimating one integral of an extremely peaked function to the problem of estimating many integrals of smoother functions. Note that simply multiplying correlated estimates of predictive probabilities does not necessarily

result in an unbiased evidence estimator. The sequential importance sampling and sequential importance resampling evidence estimators described below are unbiased, however our approach in Section 4.1 will not be.

Bayesian updating can be done one, or many observations at a time. One observation at a time is typically more computationally intensive since more operations will need to be done per data point and we cannot easily take advantage of data parallelism; however, Bayesian updating many observations at a time can result in a higher variance of the resulting estimators, since the distributions in the sequence will become more dissimilar.

Sequential importance sampling is typically used for posterior inference and evidence estimation in the context of latent variable models with data tempering, in which case each new latent variable is sampled from some importance distribution  $q_n$  which may depend on all the previous latent variables and observations. The unnormalized importance weights may be calculated for the  $i^{\text{th}}$  particle as

$$w_i^{(n)} = w_i^{(n-1)} \frac{p(y_n | z_n^{(i)}) p(z_n^{(i)} | z_{n-1}^{(i)})}{q_n(z_n^{(i)})}. \quad (3.48)$$

Here we have suppressed dependence on the model parameters for readability. The posterior expectation of a test function  $h$  can be approximated using the samples weighed by the normalized importance weights

$$\frac{\sum_i h(\theta_i, z_{1:N}^{(i)}) w_i^{(N)}}{\sum_i w_i^{(N)}}, \quad (3.49)$$

and the evidence can be approximated by the mean of the importance weights

$$\hat{\mathcal{Z}} := \frac{1}{M} \sum_i w_i^{(N)} \quad (3.50)$$

Many generalizations of this idea exist, depending on the kind of model and the dependencies between the latent variables. In any of these cases, the choice of importance distribution  $q_n$  plays a critical role in the quality of the estimator. While a simple implementation might use the prior transition distribution  $q_n(z_n) = p(z_n | z_{n-1})$ , this may result in high variance. In a more sophisticated approach,  $q_n$  could be parameterized and optimized variationally, as done in variational sequential Monte Carlo (Naesseth *et al.*, 2017).

One problem that typically arises with sequential importance sampling is degeneracy in the importance weights. This happens when the normalized importance weight for one particle is approximately one while the rest are zero. This results in an effective sample size of one and high variance of the Monte Carlo estimator. In an attempt to reduce this degeneracy, the particles can at each step be resampled with replacement from the current live particles with probability proportional to their importance weights. Resampling in this way results in particles which are distributed according to the posterior transition distribution  $p(z_n|z_{n-1}, y_{\leq n})$  in the limit of an infinite number of particles, but does not always improve the result when there are a small number of particles. Furthermore, while resampling reduces weight degeneracy, it instead results in particle degeneracy, since it kills off some of the live particles while duplicating others. To some extent this can be combated by applying a Markov transition to the resampled particles to allow them to move around; however, resampling can still result in mode collapse and so it should be used with care. The resulting algorithm is usually called sequential importance resampling (SIR).

For parametric models without latent variables, we can apply SIR with data tempering by initially sampling  $M$  particles from the prior and then, for each observation, calculating the importance weights corresponding to Bayesian updating, i.e. the ratio of the relative posterior to the relative prior, resampling the particles proportionally to these importance weights and then applying a Markov kernel which leaves the current distribution invariant. The importance weights at each step have a particularly attractive form,<sup>8</sup>

$$w_i^{(n)} = p(y_n|\theta_i^{(n-1)}). \quad (3.51)$$

Each importance weight update only depends on the current data point  $y_n$ , and does not require iterating over the entire data set. The particles are resampled proportionally to  $w_i^{(n)}$  after which each particle is updated through a Markov transition which has  $p(\theta|y_{\leq n})$  as its stationary distribution. We can estimate the posterior predictive densities for each observation

$$\hat{p}(y_n|y_{<n}) = \frac{1}{M} \sum_i w_i^{(n)}, \quad (3.52)$$

---

<sup>8</sup>The importance weights do not need to be defined recursively for SIR because the particles are weighted equally after resampling.

the product of which can be used to estimate the evidence

$$\hat{\mathcal{Z}} := \prod_n \hat{p}(y_n | y_{<n}). \quad (3.53)$$

This estimator is unbiased (Naesseth *et al.*, 2019).

Instead of resampling the particles after each observation, one can specify some resampling criterion and only resample when this criterion is met. One possible criterion could be to only resample after the ESS drops below some user-specified target. In this case the importance weights for each observation are cumulatively multiplied between resampling steps.

For an HMC kernel, the particles might be sampled approximately from the  $n^{\text{th}}$  tempered distribution using the following potential energy function:

$$U_n(\theta) = - \sum_{k \leq n} \log p(y_k | \theta) - \log p(\theta) \quad (3.54)$$

SIR with HMC for models without latent variables is summarized in Algorithm 7. We leave open the choice of resampling criterion.

---

**Algorithm 7** Sequential Importance Resampling

---

**Input:** Data  $\mathcal{D} = \{y_n\}_{n=1}^N$ , number of particles  $M$ , pdfs  $p(y|\theta), p(\theta)$

**Output:**  $\hat{\mathcal{Z}}$  evidence estimator

```

1:  $\forall_i$ : sample  $\theta_i \sim p(\theta)$ 
2:  $\forall_i$ :  $w_i \leftarrow 1$ 
3: for  $n = 1, \dots, N$  do
4:    $\forall_i$ :  $w_i \leftarrow w_i p(y_n | \theta_i)$ 
5:   if resampling criterion met then
6:     resample  $\theta_i$  proportionally to  $w_i$ 
7:      $\forall_i$ :  $w_i \leftarrow \frac{1}{M} \sum_j w_j$ 
8:   end if
9:    $\forall_i$ :  $\theta_i \leftarrow \text{HMC}(\theta_i, U_n)$   $\triangleright U_n$  defined in Equation 3.54
10: end for
11: return  $\hat{\mathcal{Z}} = \frac{1}{M} \sum_i w_i$ 

```

---

# Chapter 4

## Stochastic Gradient Evidence Estimation

This chapter presents an approach for large-scale evidence estimation using stochastic gradient MCMC algorithms. The content of this chapter is original work by the author which has recently been published. Section 4.1 covers an approach which we presented in a poster and paper submitted to the Max-Ent conference in July 2019 (Cameron *et al.*, 2019a). This work proposed estimating evidence online using Bayesian updating with a simple estimator of predictive distributions based on SG-MCMC. This approach was further developed by introducing annealing to reduce variance in the predictive probability estimates. We called this approach stochastic gradient annealed importance sampling (Cameron *et al.*, 2019b), presented in Section 4.2.

### 4.1 Sequential Evidence Estimation with SG-MCMC

Recall that evidence can be factorized into a product of predictive distributions

$$\mathcal{Z} = \prod_n p(y_n | y_{<n}), \quad (4.1)$$

where each predictive distributions takes the form of an expectation value of the current data point  $y_n$  with respect to the posterior of all previous data,

$$p(y_n | y_{<n}) = \mathbb{E}_{p(\theta | y_{<n})} [p(y_n | \theta)]. \quad (4.2)$$

This suggests that, given an estimator for predictive distributions  $\hat{p}(y_n|y_{<n})$ , the evidence could be estimated as

$$\hat{\mathcal{Z}} := \prod \hat{p}(y_n|y_{<n}). \quad (4.3)$$

Indeed this is the approach taken by SIR (Algorithm 7).

One detail that we did not fully address in our conference paper is that the estimator  $\hat{\mathcal{Z}}$  here will not necessarily be unbiased when the estimators of the predictive probabilities are correlated. However if the predictive estimators are consistent then  $\hat{\mathcal{Z}}$  will also be consistent since the bias will tend to zero as the variance of the predictive estimators tend to zero.

The main computational difficulty with this formulation is to estimate the predictive distributions in a way that scales favourably with the data set size. We can achieve linear time complexity in the resulting evidence estimation algorithm if we can guarantee constant time complexity for estimating the predictive distributions. In our conference paper, we considered the simple case of estimating predictive distributions with MCMC samples.

$$\hat{p}(y_n|y_{<n}) := \frac{1}{M} \sum_{i=1}^M p(y_n|\theta_i), \quad (4.4)$$

where the particles  $\theta_i$  are drawn approximately from the posterior  $p(\theta|y_{<n})$  using SGHMC. Since we expect each successive posterior  $p(\theta|y_{<n})$  to be similar to its relative prior  $p(\theta|y_{<n-1})$ , the particles can generally be reused and only a short burn-in time should be required to converge to the next posterior. This approach is quite similar in nature to SIR except for two key differences. Firstly, there is no resampling step, but the importance weights are still averaged upon each iteration; and secondly, we use SGHMC instead of some other MCMC method.

Averaging  $p(y_n|\theta_i)$  at each step results in some bias in the final evidence estimator since the predictive distributions are correlated, but results in a lower variance than if no averaging were done. Since we are interested in scaling to large data sets, trading some bias for reduced variance seems worthwhile. This bias would be removed if resampling had been incorporated, but we did not consider this in the conference paper.



If MH based kernels such as HMC were used instead of SGHMC, then each step of the algorithm would require iteration over all  $n$  previous observations just to generate a new state for each particle. This results in an algorithm which has at least quadratic time complexity in the data set size. Since SGHMC uses mini-batching, allowing the particles to burn in for a fixed number of iterations for each new observation results in a predictive probability estimator that has constant time complexity in the data set size, i.e. the time required to compute  $\hat{p}(y_n|y_{<n})$  does not depend on  $n$ . The resulting evidence estimator can then be computed in time linear in the data set size, which is a considerable advantage. Linear time complexity is the best that one could hope for, since calculating the probability of a data set in any reliable way necessitates visiting each data point at least once.

In order to ensure that the new data is taken into account during the SGHMC steps, we used an energy function explicitly incorporating the new data in addition to the mini-batches sampled from previous data. The potential

$$U_n(\theta) = -\log p(y_n|\theta) - \frac{n-1}{|B|} \sum_{y \in B} \log p(y|\theta) - \log p(\theta), \quad (4.5)$$

is used to generate samples approximately from the  $n^{\text{th}}$  posterior  $p(\theta|y_{\leq n})$ , where the mini-batch is sampled i.i.d. with replacement from the previously seen data  $B \subset \{y_k \mid k < n\}$ .

In practice, Bayesian updating by a single observation at a time is not particularly efficient. As  $n$  gets large, we expect that each new observation will generally contain very little new information and so by processing a number of observations at a time, for example in chunks the same size as a typical mini-batch, we can take advantage of data parallelism while requiring fewer MCMC steps in total. One difficulty we encountered is that the estimator in Equation 4.4 tends to have high variance when  $n$  is small. This high variance is worsened for high-dimensional parameter spaces and Bayesian updating with many observations at a time. The result is that when  $n$  is small, we may need to estimate predictive probabilities for a small number of observations at a time and only increase the number of observations processed at a time once  $n$  is large enough that new observations contain very little new information. For the results reported in our conference paper, we used a Bayesian updating schedule which initially only added 20 observations at a time, increasing

linearly at a rate of 20 until a maximum of 500. It is possible the schedule may need to be hand-tuned on a per-application basis and does not necessarily scale well to complex models.

Since we are interested in estimating evidence for large data sets, high variance in the initial predictive estimates may be acceptable if the final evidence estimate is still accurate. In this case the high variance in the initial terms in Equation 4.3 may be carefully corrected. To this end we further considered a hybrid approach to estimating evidence which replaces the initial high variance terms in Equation 4.3 with an estimate provided by nested sampling. This estimate can be computed accurately on a small number of data points, after which the stochastic gradient method can be applied to efficiently scale up to large data sets.

## 4.2 Stochastic Gradient Annealed Importance Sampling

Although we were able to get acceptable results using the evidence estimator discussed in the previous section, there were some clear shortcomings regarding the robustness of the approach. These shortcomings will be revisited in further detail in Section 5.3. Here, we present an algorithm we call stochastic gradient annealed importance sampling (SGAIS), which addresses these shortcomings. SGAIS is the key novel contribution presented in this thesis.

The main shortcomings of the evidence estimator described in the previous section are due to the high variance of the predictive probability estimates. Earlier predictive probability estimates exhibit high variance because the posterior is still quite diffuse and so individual observations still contain significant additional information. High variance can also ensue when some observations seem unlikely in the model, possibly due to some non-stationarity in the data which the model may not be capable of adequately describing. We can improve the quality of the final evidence estimator by instead using a more accurate approach to calculating predictive probabilities than simple averaging. For this we use AIS. This can be done by starting with  $M$  particles drawn approximately from  $\theta_i^{(n-1)} \sim p(\theta|y_{<n})$ , presumably from a previous run of AIS with

importance weights which we will label as  $w_i^{(n-1)}$ . We then treat  $p(\theta|y_{<n})$  as the prior and anneal the likelihood of new observations  $p(y_n|\theta)$ . The resulting importance weights, call them  $\tilde{w}_i^{(n)}$ , depend on the initial starting point of the particle. If the particle had been initially sampled exactly from the posterior  $p(\theta|y_{<n})$  instead of approximately, then the weights  $\tilde{w}_i^{(n)}$  would be unbiased estimates of the posterior predictive. We can instead reweight these importance weights by the corresponding importance weights for each particle produced by the previous AIS run. By using the unnormalized importance weights, we can estimate the ‘unnormalized’ posterior predictive, i.e. the probability of the total data set seen so far. Defining  $w_i^{(n)} := \tilde{w}_i^{(n)} w_i^{(n-1)}$  we have the estimator

$$\hat{p}(y_{\leq n}) := \frac{1}{M} \sum_i w_i^{(n)}. \quad (4.6)$$

The final evidence estimator is then  $\hat{\mathcal{Z}} = \frac{1}{M} \sum_i w_i^{(N)}$ . This estimator is unbiased. Define  $h(\theta_i) = \mathbb{E} [\tilde{w}_i^{(n)} | \theta_i]$ , where the expectation is taken with respect to the particle trajectory over one run of AIS starting at  $\theta_i$ . From the unbiasedness of AIS,  $\mathbb{E}_{p(\theta|y_{<n})} [h(\theta)] = p(y_n|y_{<n})$ . By linearity of expectation

$$\mathbb{E} [\hat{p}(y_{\leq n})] = \mathbb{E} \left[ \frac{1}{M} \sum_i \mathbb{E} [\tilde{w}_i^{(n)} | \theta_i^{(n-1)}] w_i^{(n-1)} \right], \quad (4.7)$$

$$= \mathbb{E} \left[ \frac{1}{M} \sum_i h(\theta_i^{(n-1)}) w_i^{(n-1)} \right], \quad (4.8)$$

$$= \mathbb{E}_{p(\theta|y_{<n})} [h(\theta)] p(y_{<n}), \quad (4.9)$$

$$= p(y_n|y_{<n}) p(y_{<n}), \quad (4.10)$$

$$= p(y_{\leq n}). \quad (4.11)$$

The third line (4.9) follows from the unbiasedness of AIS for unnormalized expectations. Thus, we no longer have to explicitly multiply the predictive probability estimates; we can just update the importance weights from the previous AIS run, and the resulting estimator will automatically take all the previous predictive probabilities into account.

This can be viewed as modifying AIS to use the sequence of intermediate distributions

$$f_n^{(\lambda)}(\theta) = p(y_n|\theta)^\lambda \left[ \prod_{k<n} p(y_k|\theta) \right] p(\theta), \quad (4.12)$$

where for each  $n$  we adaptively interpolate  $\lambda$  between zero and one. The resulting importance weight updates can be calculated without iterating over the entire data set

$$w_i^{(t)} \leftarrow w_i^{(t-1)} \frac{f_n^{(\lambda_t)}(\theta_i^{(t-1)})}{f_n^{(\lambda_{t-1})}(\theta_i^{(t-1)})} = w_i^{(t-1)} p(y_n | \theta_i^{(t-1)})^{\lambda_t - \lambda_{t-1}}. \quad (4.13)$$

Each MCMC transition then uses an SGHMC kernel with the mini-batch stochastic potential energy estimate

$$\hat{U}_n^{(\lambda)}(\theta) = -\lambda \log p(y_n | \theta) - \frac{n-1}{|B|} \sum_{y \in B} \log p(y | \theta) - \log p(\theta). \quad (4.14)$$

For combined Bayesian updating with annealing, it is important that the annealing schedules be chosen adaptively since individual observations typically contain significant information in the early stages of Bayesian updating when the posterior is still quite diffuse, but typically only contain a small amount of additional information later on once the posterior has become quite constrained. We select the annealing schedule adaptively as described in Section 3.3 where the ESS is calculated as

$$\text{ESS}(\Delta) = \frac{(\sum_i \omega_i(\Delta))^2}{\sum_i \omega_i(\Delta)^2}, \quad \omega_i(\Delta) = p(y_n | \theta_i^{t-1})^\Delta, \quad (4.15)$$

where  $\Delta := \lambda_t - \lambda_{t-1}$ .

SGAIS is summarized in Algorithm 8. Here one may include the possibility of resampling steps, similar to SIR, in which case the particles  $\theta_i$  would be resampled proportionally to their importance weights  $w_i$ , thereafter uniformly setting the importance weights to their mean  $w_i \leftarrow \frac{1}{M} \sum_j w_j$ .

### 4.3 Bayesian Updating with NS

NS requires iterating through the entire data set each time a new candidate sample is generated to ensure that each new accepted sample has a higher likelihood than the last. Because of these strict constraints, any attempt to use mini-batches violates the basic assumptions of NS. We circumvented a similar problem in AIS where the entire data set was required for the importance weights by introducing Bayesian updating, which resulted in SGAIS.

**Algorithm 8** Stochastic Gradient Annealed Importance Sampling

**Input:** Data  $\mathcal{D} = \{y_n\}_{n=1}^N$ , number of particles  $M$ , pdfs  $p(y|\theta), p(\theta)$ , target ESS: ESS\*

**Output:**  $\hat{\mathcal{Z}}$  evidence estimator

---

```

1:  $\forall_i$ : sample  $\theta_i \sim p(\theta)$ 
2:  $\forall_i$ :  $w_i \leftarrow 1$ 
3: for  $n = 1, \dots, N$  do
4:    $\lambda \leftarrow 0$ 
5:   while  $\lambda < 1$  do
6:      $\Delta \leftarrow \operatorname{argmin}_{\Delta} [\operatorname{ESS}(\Delta) - \operatorname{ESS}^*]$   $\triangleright \Delta \in (0, 1 - \lambda]$ 
7:      $\lambda \leftarrow \lambda + \Delta$ 
8:      $\forall_i$ :  $w_i \leftarrow w_i p(y_n|\theta_i)^\Delta$ 
9:     optionally resample particles
10:     $\forall_i$ :  $\theta_i \leftarrow \operatorname{SGHMC}(\theta_i, \hat{U}_n^{(\lambda)})$   $\triangleright \hat{U}_n^{(\lambda)}$  defined in Equation 4.14
11:   end while
12: end for
13: return  $\hat{\mathcal{Z}} = \frac{1}{M} \sum_i w_i$ 

```

---

It therefore seems worthwhile to investigate Bayesian updating in the context of NS. For the experiments in [Cameron \*et al.\* \(2019a,b\)](#) we implemented NS with an SGHMC kernel to sample from the constrained prior. The constrained sampler used momentum reflection to reflect off of the iso-likelihood contours similarly to Galilean Monte Carlo as described in Section 3.2. With this implementation already complete it was trivial to apply NS to estimating predictive probabilities, treating  $p(\theta|y_{<n})$  as the prior and applying SGHMC to update particles in constant time complexity in  $n$ . However for each run of NS, we need particles which are initially sampled approximately according to  $p(\theta|y_{<n})$ . NS generates samples as a by-product, but they are not distributed according to the posterior. Instead, one can generate approximate posterior samples by resampling the points which were generated by NS proportionally to their weight

$$\frac{\lambda_k w_k}{\hat{\mathcal{Z}}}, \quad (4.16)$$

where  $\lambda_k, w_k$  and  $\hat{\mathcal{Z}}$  are as described in Section 3.2 — see Equation 3.26. Starting with these resampled particles, each subsequent run of NS produces an estimate of the predictive probability  $p(y_n|y_{<n})$  which is consistent in the limit of infinite number of particles.

One challenge for computational efficiency is resampling the particles without

keeping a buffer of all the particles sampled during each subsequent run of NS. This can be achieved through weighted reservoir sampling ([Chao, 1982](#)), allowing one to keep a buffer of only size  $M$  when  $M$  particles are used.

Unfortunately, preliminary experiments indicated that this approach would not give any performance gain over vanilla NS for a single, fixed size data set.<sup>1</sup> However, when used in an online setting, this approach still has the benefit of utilizing previous evidence estimates, resulting in efficient marginal updates. Preliminary tests seemed to indicate that our online version of NS was able to produce evidence estimates on large data sets in time similar to vanilla NS, but with the potential added benefit of giving an estimate of the evidence for each smaller data set  $\{y_k\}_{k=1}^n$  for various values of  $n$ .

---

<sup>1</sup>There is some possibility that this approach could have improved performance using some carefully selected adaptive convergence criterion; however, it is not immediately obvious how this could be done in a reliable way.

# Chapter 5

## Simulation Results

This chapter reports the experiments and results presented in [Cameron \*et al.\* \(2019a,b\)](#). A significant amount of text as well as figures have been taken verbatim from these papers.

### 5.1 Methodology

To evaluate the accuracy and runtime performance of our proposed approach, we estimated the evidence for three simple models on simulated data sets in an online fashion. We further evaluated the robustness of SGAIS under various choices of algorithm parameter values.

As already set out in [Section 3.1](#) The log-evidence typically grows linearly in the number of data points. For this reason, it is natural to measure errors in  $\log \mathcal{Z}/N$  rather than  $\log \mathcal{Z}$ . For each model in [Section 5.2](#), we measured the runtime performance of SGAIS compared to NS and AIS for various data set sizes up to one million observations. Each of these “interim” data sets is taken as the first  $N$  observations of the largest data set.

#### 5.1.1 Default Parameters

We use mini-batches of size 500 with the SGHMC parameters set to  $\eta = 0.1/N$ ,  $\alpha = 0.2$ , and  $\hat{\beta} = 0$ . Predictive distributions are approximated using  $M = 10$  particles and 20 burn-in steps for each intermediate distribution. We use a target ESS of 5 for adaptive annealing. Rather than Bayesian updating

by adding a single observation at a time, we add chunks of data at a time that are the same size as the mini-batches. These parameter choices are not necessarily the optimal choice for the models below. Instead, we chose the SGHMC parameters based approximately on those suggested in [Chen \*et al.\* \(2014\)](#), and we chose the number of particles and target ESS to hopefully be sufficient for adaptive annealing but small enough to result in a short running time.

### 5.1.2 NS and AIS

As our reference standards of accuracy, we implemented NS and AIS. Both NS and AIS implementations used SGHMC as their MCMC kernel. For AIS, we used the same parameters as SGAIS, except that each MCMC step uses the whole data set instead of mini-batches. We implemented NS with 20 SGHMC steps to sample from the constrained prior. NS still requires the full data set to check the constraints and so cannot take advantage of mini-batching. For SGHMC used with NS, we used parameters  $\eta = 10^{-3}$ ,  $\alpha = 0.1$ , and  $\hat{\beta} = 0$  because there is no gradient noise when sampling from the prior. Results reported are for two particles; more particles result in similar but slower behaviour. We allow NS to run until the additive terms are less than 1% of the current  $\hat{\mathcal{Z}}$  estimate. This is a popular stopping criterion and is also used in [Grosse \*et al.\* \(2015\)](#). Since we found AIS to be much slower than NS, we only ran AIS on data sets small enough to finish within 4000 seconds.

The results from our conference paper, shown in Section 5.3 only compare to NS, since we had not yet implemented any annealing algorithms, while the results in Section 5.4 compare SGAIS to both NS and AIS. We consider discrepancies between results from different algorithms to be acceptable if they are of the same magnitude as discrepancies between NS and AIS. Since all these estimators are likely to underestimate the evidence, it is generally safe to assume the largest evidence estimate is the most accurate.

### 5.1.3 Sensitivity Analysis

We investigated the sensitivity of SGAIS to the following parameters: number of particles  $M$ , the target ESS, the number of burn-in steps for each intermediate distribution, the learning rate  $\eta$ , and the mini-batch size  $|B|$ . Each



test was done by varying one parameter while keeping the others fixed at their default values.

#### 5.1.4 Run-time Environment

Our experiments were executed on a laptop with an Intel i7 CPU and 8GB of RAM, running Arch Linux; kernel release 5.1.7. For fair comparison, all code was single-threaded. Multithreading gives a considerable speedup when calculating the likelihood on large data sets but can introduce subtle complexities that are difficult to control and quantify in tests of run-time performance.

#### 5.1.5 Non-stationarity Detection

During online estimation, changes in the data-generating distribution should typically be detectable in the evidence estimates. To investigate this, we generated simulated data with varying numbers of “clusters” of data points based on different simulation parameter values. Histograms of the simulated data are shown in Figure 5.1. The first 1000 observations were generated from 3 Gaussian distributions, the next 9000 observations were generated from 5 Gaussian distributions, including the 3 used to generate the first observations, and the remaining 90,000 observations were generated from 7 Gaussian distributions, including the previous 5. Some of the clusters overlap, so it is not immediately obvious from the histograms how many clusters there actually are.

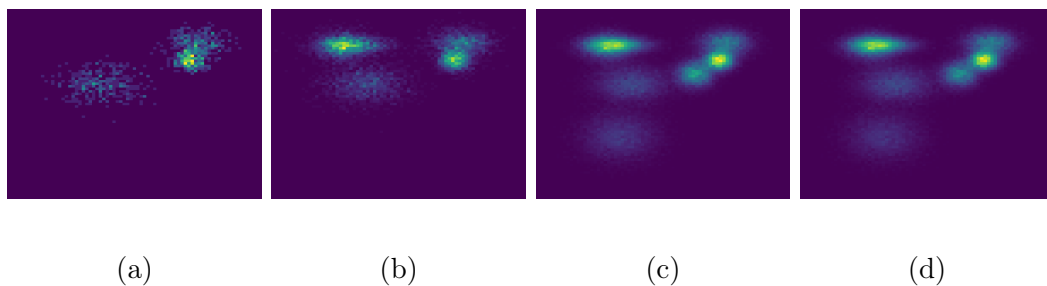


Figure 5.1: Histograms of non-stationary simulated data. (a) shows the first 1000 observations, (b) shows the next 9000 observations, (c) shows the last 90,000 observations, and (d) shows the total data set. In each of the three time phases, the data-generating distribution produces data with more clusters than before.

We evaluated the effect of this type of distribution shift on SGAIS by estimating the evidence online for Gaussian mixture models — see Section 5.2 — with 3, 5, and 7 mixture components. For comparison, we then shuffled the data to enforce stationarity and estimated the evidence for these three models on the shuffled data. If the final evidence estimates for the in-order and shuffled data differ significantly then this may indicate that the particles are getting trapped in local modes before the change-points occur.

## 5.2 Models

Our experiments were performed on the following three models using data generated by sampling from the model’s conditional distributions.

### 5.2.1 Linear Regression

The data set consists of pairs  $(x, y)$  where  $x$  is a vector and  $y$  is a scalar related by

$$y = w^T x + b + \epsilon,$$

where  $\epsilon$  is zero mean Gaussian distributed with known variance  $\sigma^2$ . We do not assume any distribution over  $x$  as it always appears on the right-hand side of the conditional. The single-observation likelihood is

$$p(y|x, w, b) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - w^T x - b)^2}{2\sigma^2}\right).$$

Parameters are  $w$  and  $b$ , with standard Gaussian priors. Evidence can be calculated analytically for this model. We used 5 dimensional vectors  $x$ ; this model therefore has 6 parameters.

### 5.2.2 Logistic Regression

The data set consists of pairs  $(x, y)$ , where  $x$  is an observation vector that is assigned a class label  $y \in \{1, \dots, K\}$ . The labels have a discrete distribution with probabilities given by the softmax function of an affine transform of the observations

$$p(y|x, \theta) = \frac{\exp(w_y^T x + b_y)}{\sum_k \exp(w_k^T x + b_k)}.$$

Parameters are  $\theta = (w_{1:K}, b_{1:K})$ , with standard Gaussian priors. Again, we do not assume any distribution over  $x$  as it always appears on the right-hand side of the conditional. We used 10 dimensional vectors  $x$  with 4 classes; this model has 44 parameters.

### 5.2.3 Gaussian Mixture Model

The data are modeled by a mixture of  $d$ -dimensional multivariate Gaussian distributions with diagonal covariance matrices. Mixture weights, means, and variances are treated as parameters. This type of model is often treated as a latent variable model, where the mixture component assignments of each data point are the latent variables. Here, we marginalize out the latent variables to obtain the following conditional distribution:

$$p(y|\theta) = \sum_{k=1}^K \beta_k \prod_{j=1}^d \frac{1}{\sqrt{2\pi\sigma_{k,j}^2}} \exp\left(-\frac{(y_j - \mu_{k,j})^2}{2\sigma_{k,j}^2}\right),$$

with

$$\theta = (\beta_{1:K}, \mu_{1:K,1:d}, \sigma_{1:K,1:d}^2).$$

Mixture weights  $\beta_{1:K}$  are modeled by a Dirichlet prior with  $\alpha = 1$ ; means  $\mu_{k,j}$  are modeled conditionally given the variances by Gaussian priors, centered around zero with variance  $4\sigma_{k,j}^2$ ; variances  $\sigma_{k,j}^2$  are modeled by inverse gamma priors with shape and scale parameters equal to 1. We used 5 Gaussian components, and observations were 2-dimensional. This model has 25 parameters with 24 degrees of freedom. We use this model for our sensitivity tests, since it has the most complex structure.

## 5.3 Sequential Estimation with SG-MCMC

Here we show the results of our sequential approach which appeared in [Cameron \*et al.\* \(2019a\)](#). Our experiments here compare the approach described in Section 4.1 to NS for the three models given in Section 5.2. In some of the figures below, the sequential sampler initially underestimates the log-evidence. We believe this is due to higher variance of the initial predictive estimates when  $n$  is small, and so we also give a hybrid result which replaces the initial terms in the log-evidence estimator with an NS estimate of the evidence for the first 100 data points.

### 5.3.1 Linear Regression

For the linear regression model, the exact evidence is available analytically and is shown in Figure 5.2a for comparison. Both algorithms are able to produce accurate results for this model for all data set sizes. The final error of the sequential sampler on one million data points is only about  $10^{-4}N$  (roughly 0.01%). For this model, our method was faster than NS by about a factor of 3 on one million observations.

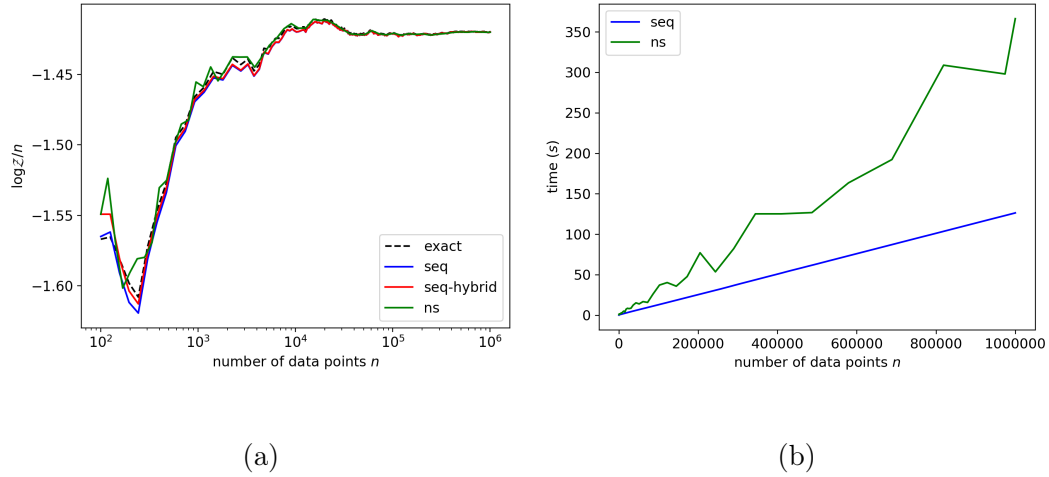


Figure 5.2: Linear regression model. (a) shows the accuracy of the sequential evidence estimator (**seq**) compared to nested sampling (**ns**) and the exact evidence as well as the hybrid approach (**seq-hybrid**). (b) shows their run-times.

### 5.3.2 Logistic Regression

Results for the logistic regression model are shown in Figures 5.3. For the largest data set, NS and our sequential sampler produced estimates which differed by  $3 \times 10^{-4}N$  (roughly 0.7%), which is negligible. Our sequential sampler was almost a factor 17 faster than the nested sampler on one million observations for this model.

### 5.3.3 Gaussian Mixture Model

The posterior distribution for this model is multimodal. Some modes are due to permutation symmetries; these modes do not have to be explored since each

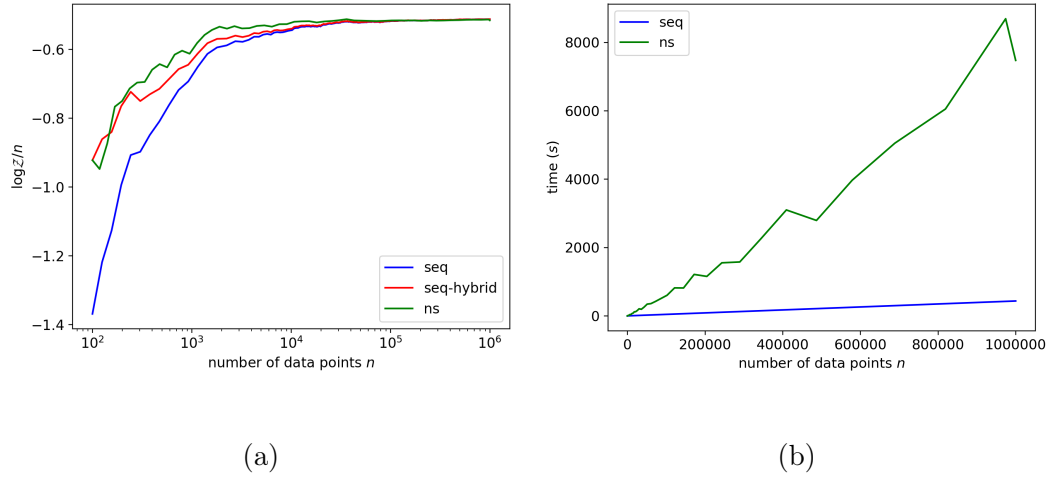


Figure 5.3: Logistic regression model. (a) shows the sequential evidence estimator compared to nested sampling and (b) shows their run-times.

one contains the same information. There are also some local modes which do not necessarily capture meaningful information about the data; for example, fitting a single Gaussian to the whole data set may be a local optimum of the likelihood function, but a poor one. If an MCMC walker finds one of these modes it can get trapped. However, we found that by Bayesian updating, the MCMC walkers tend to leave the poor local modes early on, before they become extremely peaked. This is similar to how annealing can help prevent MCMC and optimization algorithms from getting trapped in poor local optima.

Results for the Gaussian mixture model are shown in Figure 5.4. The estimates produced by NS and our sequential sampler differed on the largest data set by  $2 \times 10^{-3}N$  (roughly 0.06%). For this model our sequential sampler was about a factor 11 faster than the nested sampler on one million observations.

In all the experiments our sequential sampler seems to converge to the same result as NS within a negligible error for large data sets. The initial disagreement between NS and our sequential sampler on the first few thousand data points, as seen in Figure 5.3a and Figure 5.4a, may safely be attributed to high variance in the early stages of Bayesian updating, since the proposed hybrid approach, replacing early terms in the sequential estimator by estimates based on NS, matches NS closely for all data set sizes.

This discrepancy is a clear shortfall of our approach; it served to motivate the introduction of annealing steps, leading to the SGAIS for which results are

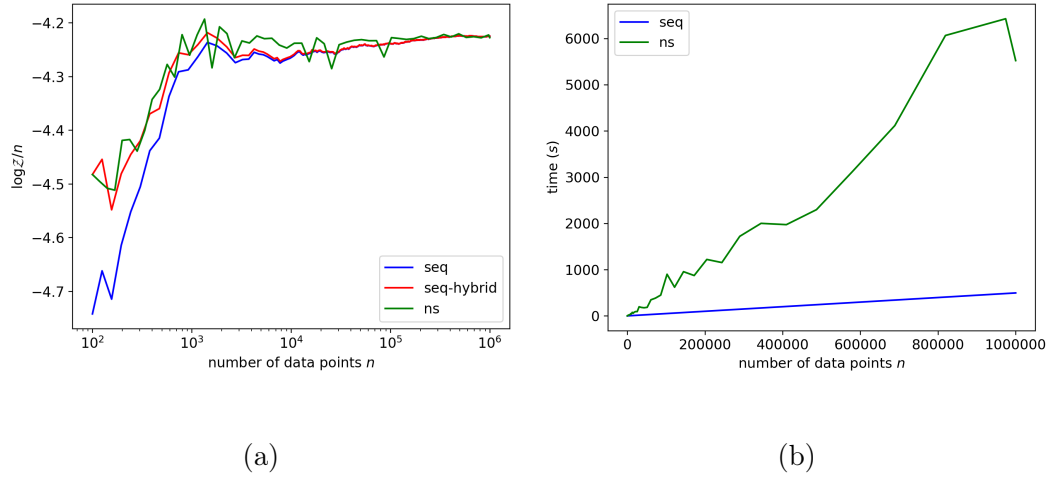


Figure 5.4: Gaussian mixture model. (a) shows the sequential evidence estimator compared to nested sampling and (b) shows their run-times.

given in the next section.

## 5.4 Results For SGAIS

The results given in this section appeared in our paper in the journal *Entropy* (Cameron *et al.*, 2019b). All models and parameter values remain the same as in Sections 5.1 and 5.2.

### 5.4.1 Accuracy and Speed

#### Linear Regression

Results are shown in Figure 5.5. The final discrepancy of SGAIS on one million data points was only about 0.1%. For this model, our method achieved a speedup over NS by about a factor of 3.3, and a speedup over AIS by a factor of 24.9 on one million observations.

#### Logistic Regression

Figure 5.6 shows the log-evidence estimates and run-time of each algorithm for the logistic regression model. For the largest data set, NS and SGAIS produced estimates that differed by roughly 0.6%, which is negligible. SGAIS was a factor 10.4 faster than the nested sampler on one million observations for

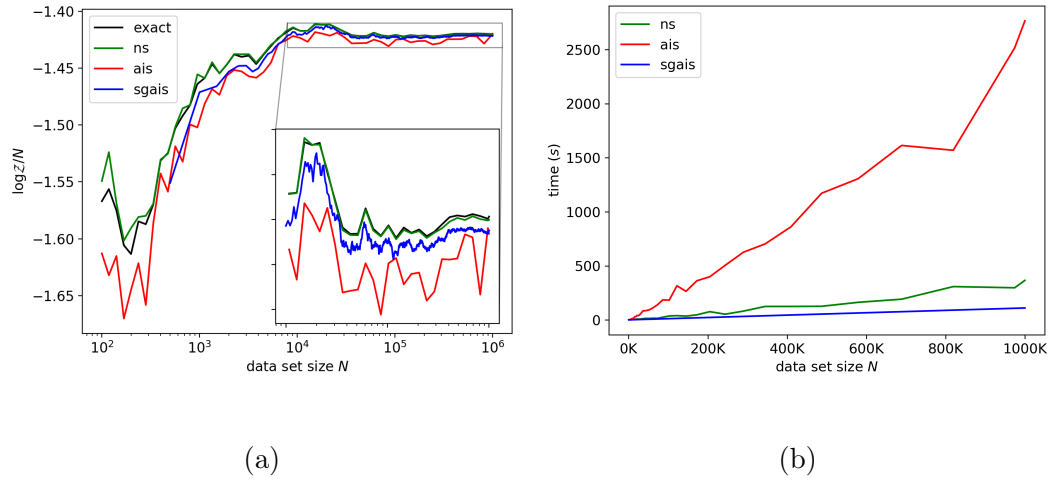


Figure 5.5: Linear regression model. (a) shows the accuracy of SGAIS estimator compared to NS, AIS, and the exact evidence. (b) shows the run-time of each method. **ns** is nested sampling, **ais** is annealed importance sampling, and **sgais** is our stochastic gradient annealed importance sampling approach.

this model. AIS was not run for larger data set sizes because each subsequent run would take more than 4000 seconds.

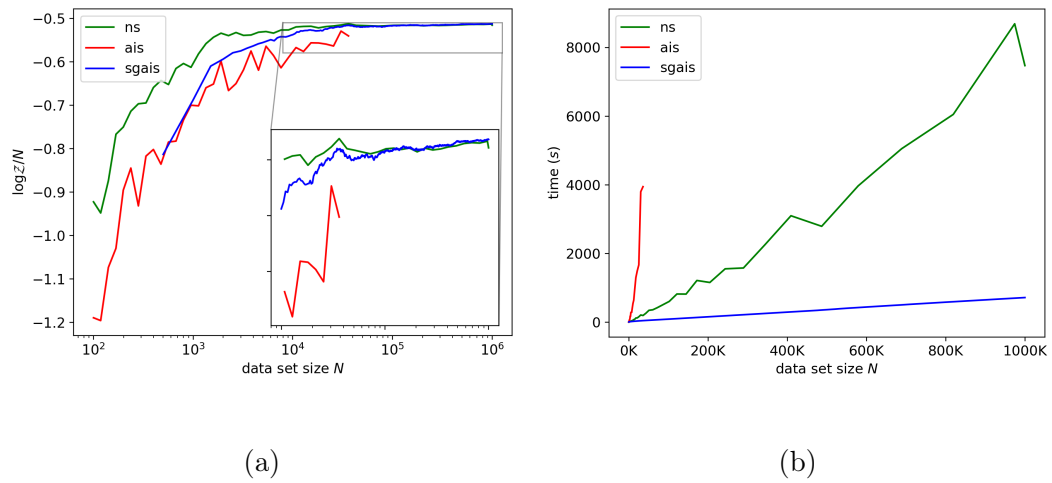


Figure 5.6: Logistic regression model. (a) shows SGAIS compared to NS and AIS. (b) shows the run-time of each method.

## Gaussian Mixture Model

Figure 5.7 shows the log-evidence estimates and run-time of each algorithm for the Gaussian mixture model. The estimates produced by NS and SGAIS

differed on the largest data set by roughly 0.1%. For this model, SGAIS was about a factor of 4.9 faster than the nested sampler for one million observations.

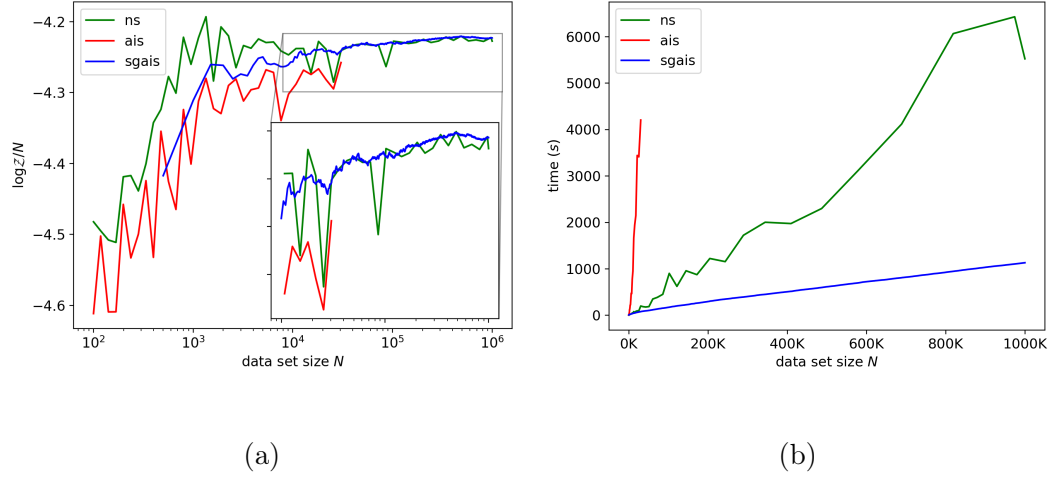


Figure 5.7: Gaussian mixture model. (a) shows SGAIS compared to NS and AIS. (b) shows the run-time of each method.

In all the above experiments, SGAIS converges to the same result as NS with a negligible error for large  $N$ . The speedup of SGAIS over NS was reduced in comparison to the sequential estimator results in Section 5.3; however, SGAIS appears to be significantly more reliable, producing accurate estimates for all data set sizes, falling within the discrepancy band bounded by AIS and NS even for small  $N$ .

Furthermore, SGAIS produces evidence estimates for all values of  $N$  in a single run, while NS and AIS only produce estimates for a single value of  $N$ . The times required to generate the above plots for NS and AIS are therefore the total area under the curves shown in Figures 5.5b, 5.6b and 5.7b.

### 5.4.2 Non-stationarity Detection

The log-evidence estimates shown in Figure 5.8a display sharp changes at 1000 and 10,000 observations for the non-shuffled data. The cusps in the resulting plot clearly identify the position of the change-points, without a priori assuming the existence or number of change-points. The numbers of annealing steps shown in 5.8b exhibit spikes at the change-points, and remain high once more



clusters are added to the data than the model can describe. The agreement

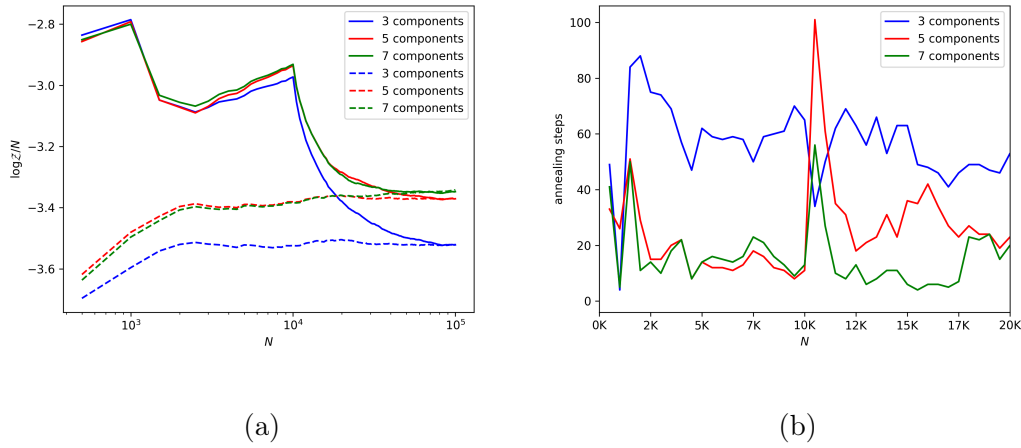


Figure 5.8: Evidence estimation under distribution shift. (a) shows the evidence estimates for Gaussian mixture models with different numbers of mixture components. The solid lines are for the non-stationary data in its original order, while the dashed lines are for the shuffled and therefore stationary data. (b) shows the number of annealing steps for the online evidence estimates for the in-order data.

of the final evidence estimates between the shuffled and non-shuffled data suggests that these estimates can be trusted. The difference between the online and shuffled estimates is small enough to be able to distinguish between the three models. The 5- and 7-component models seem to describe the total data set better than the 3-component model, but the 5- and 7-component models have similar values for their log-evidence, presumably due to the overlapping clusters in the data set.

### 5.4.3 Sensitivity to Algorithm Parameters

To evaluate the robustness of SGAIS, we also tested its dependence on parameters with reasonably predictable influences.

#### Number of Particles

Increasing the number of particles,  $M$ , results in higher accuracy and a longer running time without much effect on the number of annealing steps.

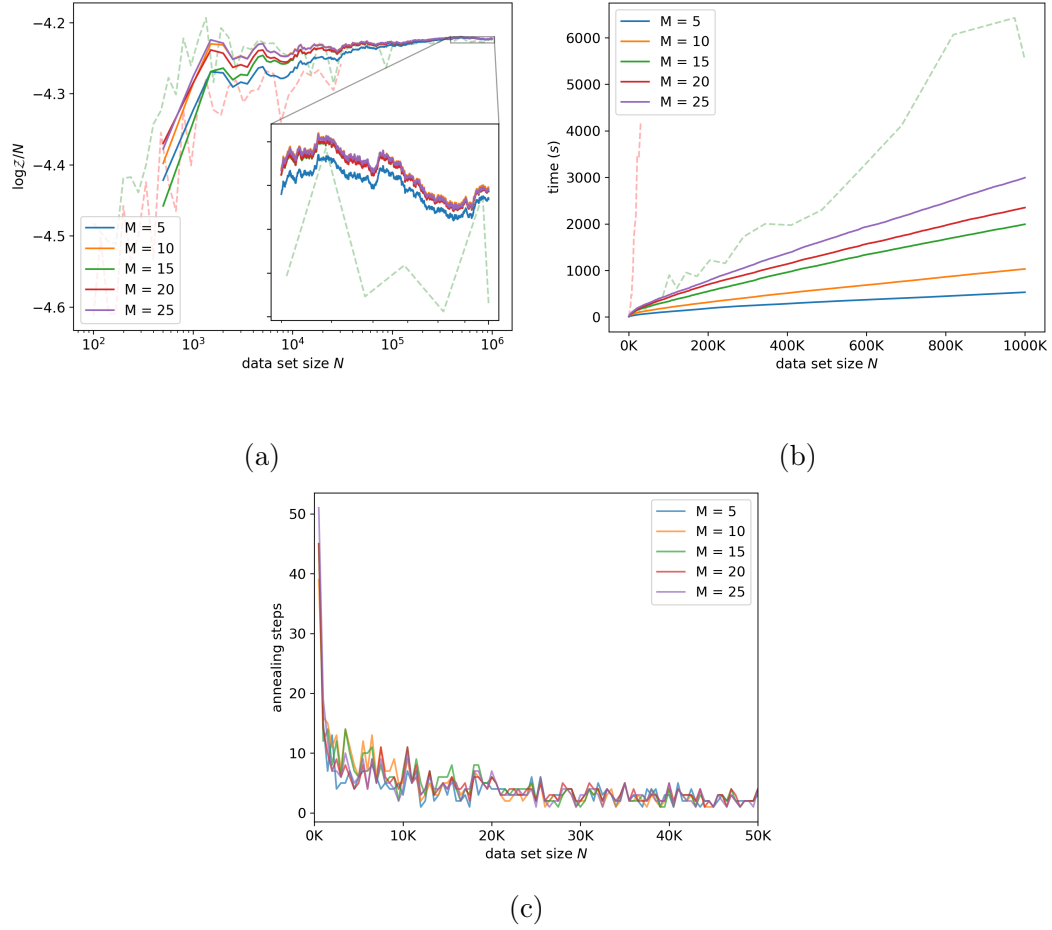


Figure 5.9: Sensitivity to the number of particles ( $M$ ). (a) shows the log-evidence estimates, (b) shows the run-time, and (c) plots the number of annealing steps for each chunk of data against the data set size until that chunk.

### Burn-in Steps

A smaller number of burn-in SGHMC steps per intermediate distribution typically resulted in lower accuracy and a shorter run-time. A smaller number of burn-in steps also resulted in more annealing steps due to the slower equilibration.

### Mini-batch Size

Larger mini-batch sizes typically result in higher accuracy but more computation per SGHMC step. Larger mini-batch sizes result in fewer Bayesian updating steps but require more annealing steps per new chunk of data. Mini-batch size would typically be chosen based on the hardware capabilities of the

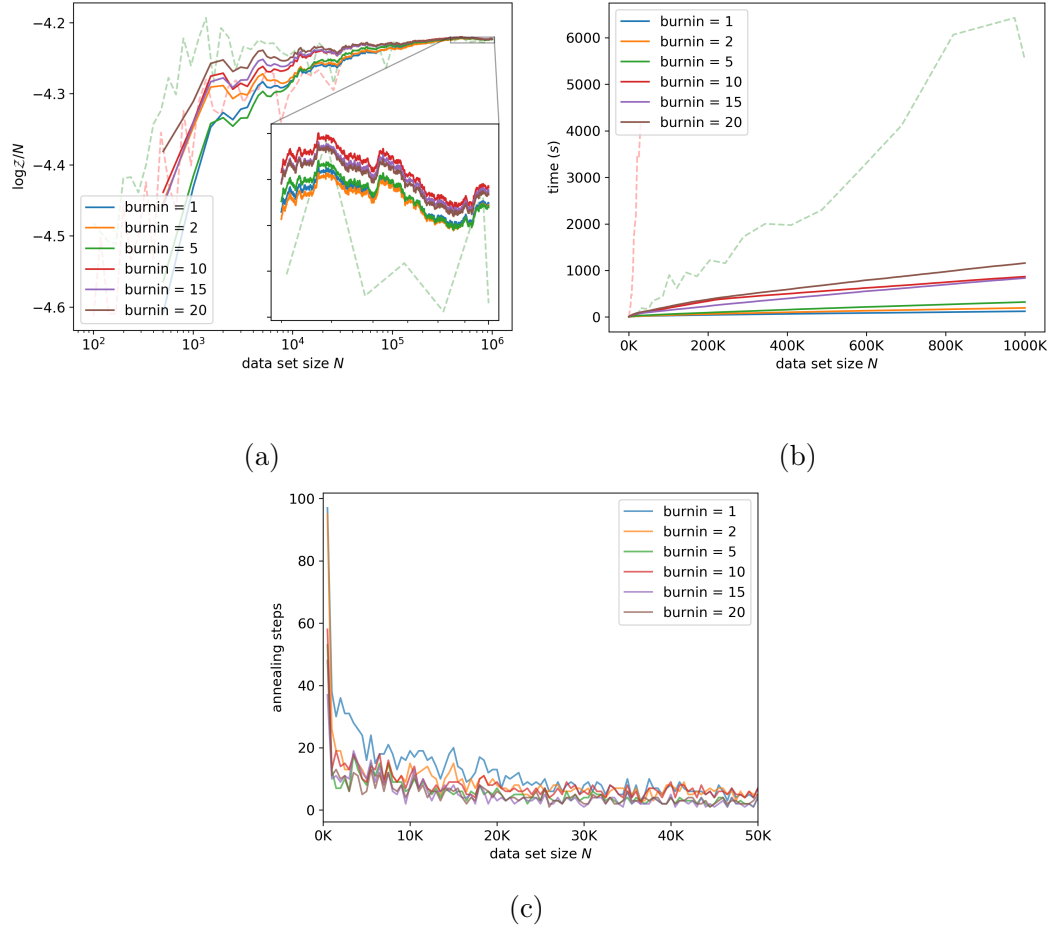


Figure 5.10: Sensitivity to the number of SGHMC burn-in steps. (a) shows the log-evidence estimates, (b) shows the run-time, and (c) plots the number of annealing steps.

platform and the type of data under consideration.

### Target ESS

As expected, a higher target ESS tends to result in more annealing steps—see Figure 5.12a. Most of the annealing work is done in the early stages of Bayesian updating. Note that since we used 10 particles, a target ESS of  $0.1M = 1$  requires no annealing steps because ESS is bounded below by 1. No annealing results in high variance during the early stages of Bayesian updating, and adaptively annealing helps to reduce that variance, with only a small impact on the run-time. This illustrates the importance of the adaptive annealing schedule in our approach. Figures 5.12a and 5.12b indicate that

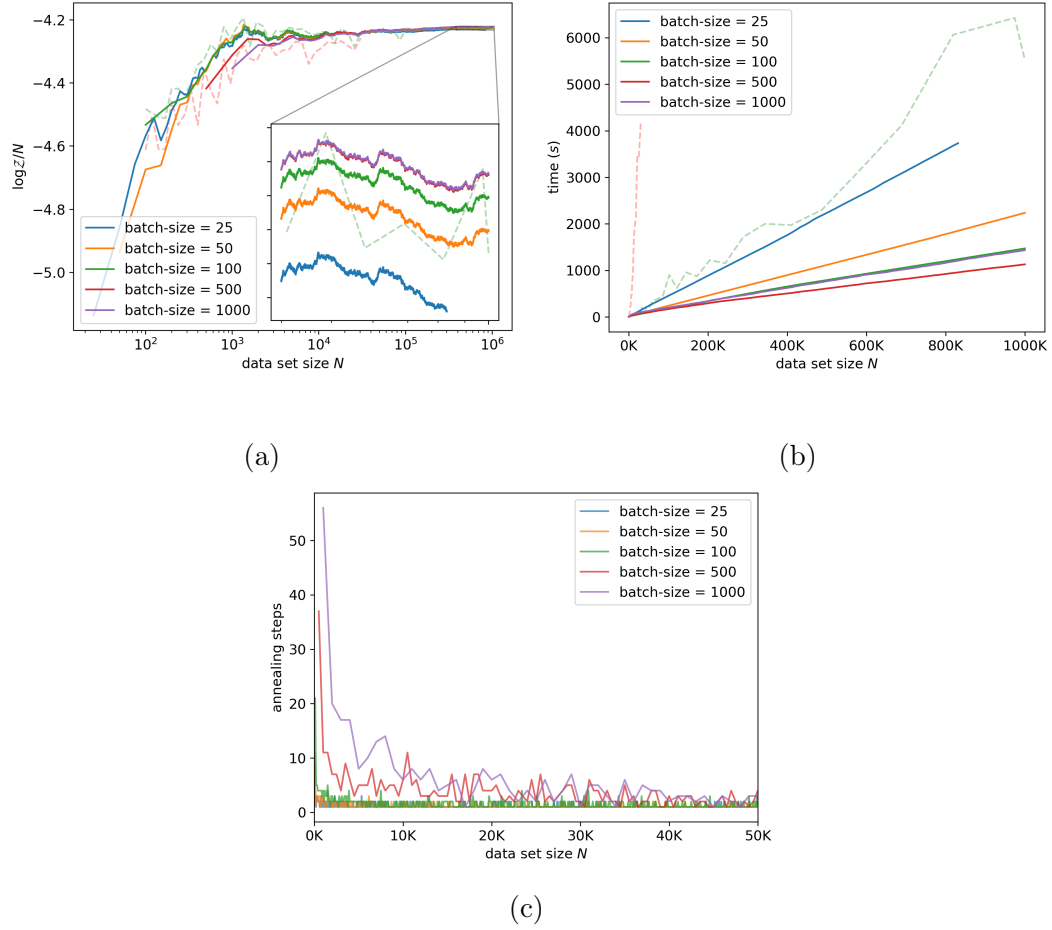


Figure 5.11: Sensitivity to the mini-batch size. (a) shows the log-evidence estimates, (b) shows the run-time, and (c) plots the number of annealing steps.

our approach converges to the log-evidence with acceptable accuracy within a reasonable time for a target ESS larger than 1. Even a small target ESS was good enough to match vanilla AIS, on average.

### Learning Rate

Interestingly, smaller values of the learning rate tend to result in less accurate log-evidence estimates over a longer time — see Figure 5.13a and 5.13b. We suspect this to be because a smaller learning rate does not allow the particle to move as far each step, resulting in a slower equilibration and requiring more annealing steps per observation. This effect can be seen in Figure 5.13c; the smaller learning rates appear to result in a larger number of annealing steps per observation. To further verify this, we investigated the interaction between

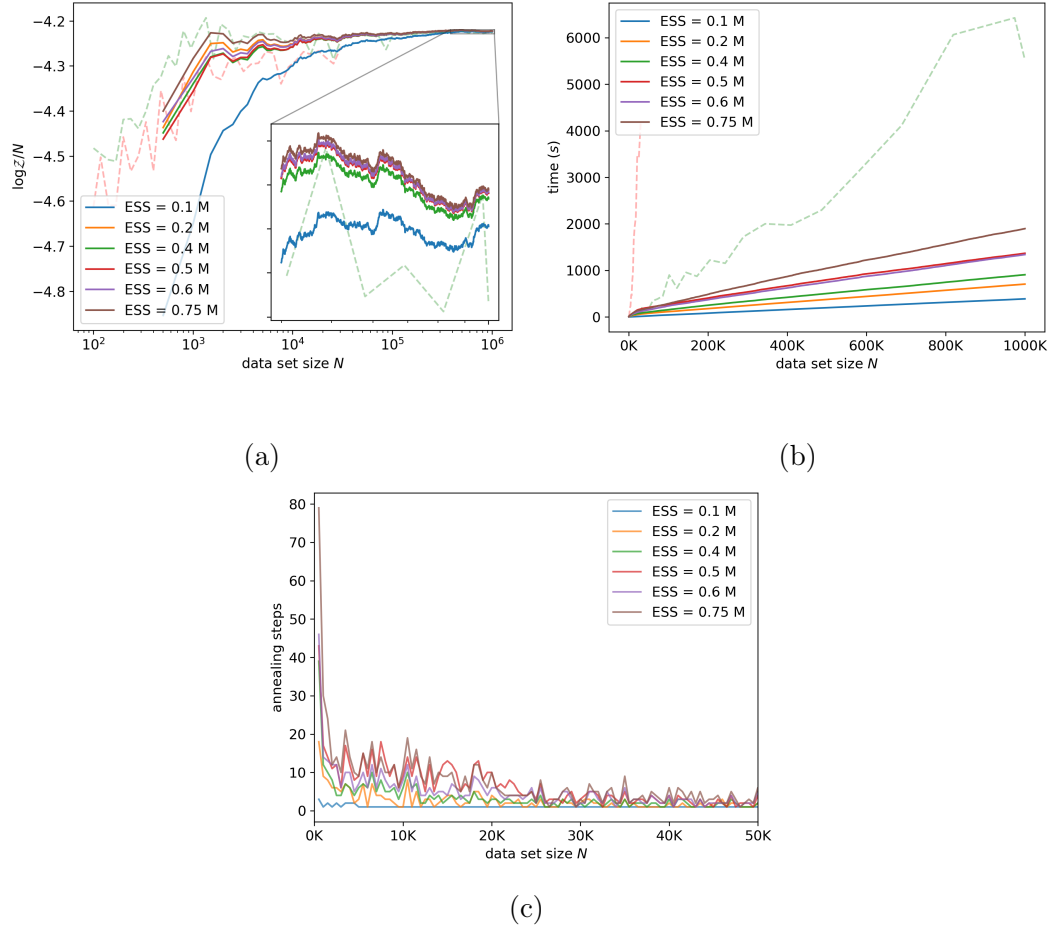


Figure 5.12: Sensitivity to the target effective sample size (ESS). (a) shows the log-evidence estimates, (b) shows the run-time, and (c) plots the number of annealing steps.

the learning rate and the number of burn-in steps.

### Learning Rate and Burn-in

We investigate the interaction between the number of SGHMC steps taken per intermediate distribution and the learning rate by varying the learning rate, while keeping the product of the learning rate and the number of SGHMC steps constant. While fewer burn-in steps (larger learning rate) tends to make the algorithm faster, a smaller learning rate results in higher accuracy in the log-evidence estimates, as seen in Figure 5.14a. The decrease in accuracy with a larger learning rate is presumably due to the discretization error in the Euler–Maruyama integrator. This conjecture is supported by Figure 5.14c: a

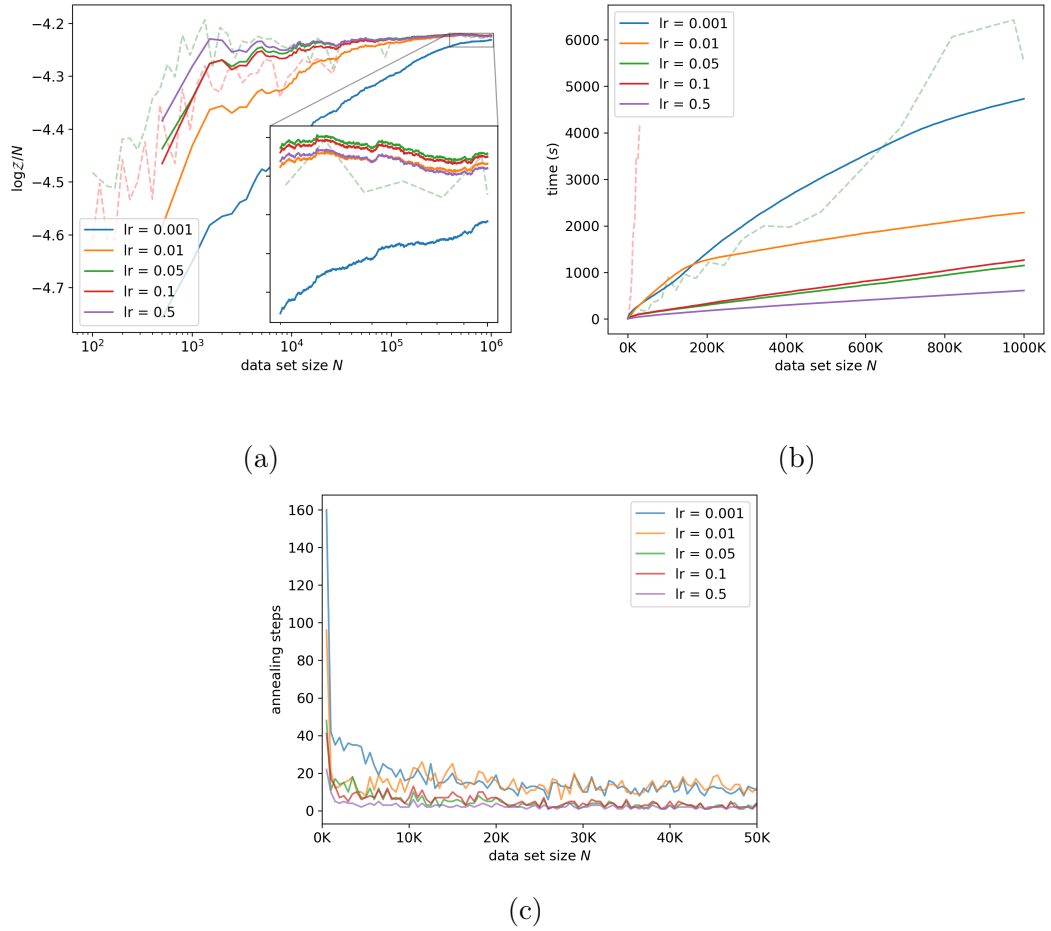


Figure 5.13: Sensitivity to the learning rate. Reported learning rates are per-observation learning rates, that is,  $\eta = lr/N$ . (a) shows the log-evidence estimates, (b) shows the run-time, and (c) shows the number of annealing steps. For a learning rate of 0.01, the run-time shown in (b) displays a change in gradient near  $10^5$  observations. This is the result of a reduced number of annealing steps but is not visible in (c) since we only show the number of annealing steps for up to  $5 \times 10^4$  observations.

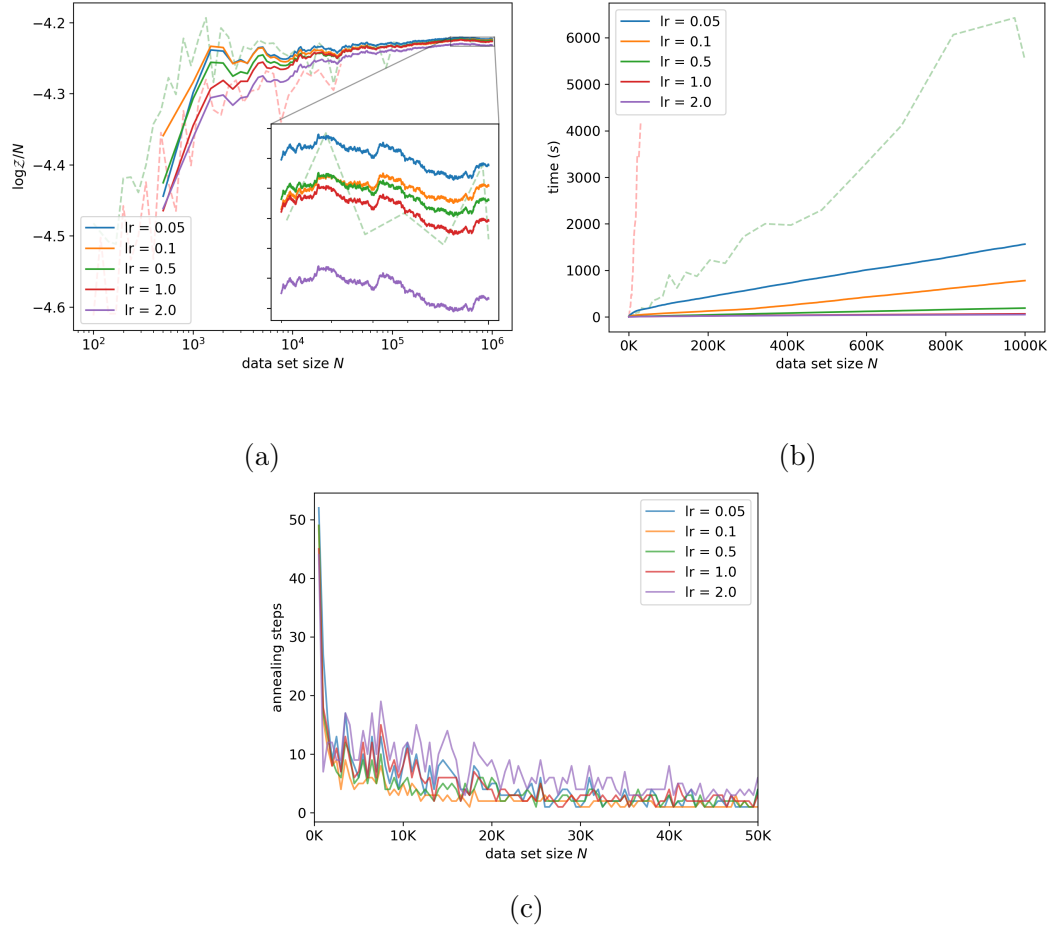


Figure 5.14: Sensitivity to the learning rate while keeping the product of the learning rate and the number of stochastic gradient Hamiltonian Monte Carlo (SGHMC) steps constant. Reported learning rates are per-observation learning rates, that is,  $\eta = lr/N$ . (a) shows the log-evidence estimates, (b) shows the run time, and (c) shows the number of annealing steps.

larger learning rate requires a larger number of annealing steps to reach the target ESS. Figure 5.14a indicates that a per-observation learning rate of 1.0 can be used and still result in estimates of acceptable accuracy on data sets of one million observations. For a learning rate of 1.0, SGAIS achieved a speedup over NS by a factor of 83.

# Chapter 6

## Conclusions

In this thesis we considered the problem of estimating Bayesian evidence in the large data regime. Bayesian evidence is useful in many applications, notably for model comparison, model combination and change-point detection. The main inhibiting factor for scaling up many previously existing algorithms originates with their need to repeatedly compute the exact likelihood over the whole data set. We therefore tackled this problem by introducing mini-batching and stochastic gradients into evidence estimation algorithms, culminating in stochastic gradient annealed importance sampling (SGAIS). SGAIS combines stochastic gradient Markov chain Monte Carlo and annealed importance sampling with mini-batch Bayesian updating and is therefore naturally applicable to online evidence estimation, even when the data exhibits non-stationarities which the model cannot account for. This approach enables efficient large-scale evidence estimation in a variety of contexts.

### 6.1 Findings

- We empirically evaluated the accuracy and performance of SGAIS compared to nested sampling (NS) and vanilla annealed importance sampling (AIS) and were able to achieve significant speedups over NS and AIS on data sets of up to one million observations for three simple models.
- We evaluated the sensitivity of our approach to the algorithm parameters and found that it was robust to a large range of parameter values within



reason.

- We further proposed an experiment to test how well SGAIS is able to detect data non-stationarities when applied in an online setting and found that we were able to very precisely identify change-points in the evidence estimates as well as in the number of annealing steps required during Bayesian updating. The evidence estimates for the in-order data agreed very precisely with estimates on the same data after it had been shuffled, indicating that estimates for online and possible non-stationary data are reliable.

## 6.2 Limitations

As with any algorithm based off of local updates, our approach is susceptible to the usual problems encountered in exploring multimodal distributions. This is due to the extremely low chance for an MCMC walker to jump from one mode to the next. There exist approaches to reduce this limitation, the simplest of which is simply adding many more particles; however, to our knowledge, there is no method which completely mitigates this problem.

SGAIS, similarly to AIS, does not provide any obvious way to approximate the accuracy of the evidence estimates. Given the extremely peaked nature of the likelihood in general and the tendency of evidence estimators to underestimate the exact evidence, sample variances of many independent runs of SGAIS cannot provide reliable uncertainty estimates. Preliminary sandwiching estimates as described in Section 3.1 and Grosse *et al.* (2015) can provide some approximation of the accuracy under certain assumptions, but this is not always possible in practice.

In order to make use of stochastic gradients and mini-batching, we have throughout this thesis assumed that the likelihood admits a certain factorization. While this is true for many models widely used in practice, the result is that SGAIS is not as universally applicable as NS or AIS. In particular, NS and AIS can be used to marginalize out hyper parameters in Gaussian process models, while SGAIS cannot.

It speaks for itself that the conclusions reached so far have been tested only

for the three simulation models described. While we are confident that the SGAIS method will perform in a wider context, this remains to be verified in practice. In particular, it is unclear how accurate the evidence estimates produced by SGAIS would be for very high dimensional parameter spaces.

## 6.3 Future Work

Many sequential Monte Carlo algorithms and online estimation applications involve latent variable models. We briefly mentioned these in Section 3.4 but did not allow for the possibility of latent variables in our approaches in Chapter 4. Possible future work may introduce latent variables and extend the framework to stochastic gradient sequential Monte Carlo. Latent variables could be estimated with importance sampling which might be optimized variationally as in Naesseth *et al.* (2017).

These methods may further incorporate other innovations from the SG-MCMC literature such as Riemannian manifold methods by estimating the local curvature, or improving equilibration times with a Nosé–Hoover thermostat.

# List of References

- Arnold, V. (1989). *Mathematical methods of classical mechanics*, vol. 60. Springer. (Cited on pages [12](#) and [13](#).)
- Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press, New York, NY, USA. ISBN 0521518148, 9780521518147. (Cited on page [3](#).)
- Beskos, A., Jasra, A., Kantas, N. and Thiery, A. (2016 04). On the convergence of adaptive sequential monte carlo methods. *Ann. Appl. Probab.*, vol. 26, no. 2, pp. 1111–1146.  
Available at: <https://doi.org/10.1214/15-AAP1113> (Cited on page [47](#).)
- Brooks, S., Gelman, A., Jones, G. and Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. CRC press. (Cited on pages [10](#), [15](#), and [18](#).)
- Brumfiel, G. (2011). High-energy physics: Down the petabyte highway. *Nature*, vol. 469, no. 7330, pp. 282–283. ISSN 1476-4687.  
Available at: <https://doi.org/10.1038/469282a> (Cited on page [4](#).)
- Buchholz, A., Chopin, N. and Jacob, P.E. (2018 Aug). Adaptive Tuning Of Hamiltonian Monte Carlo Within Sequential Monte Carlo. *arXiv e-prints*, p. arXiv:1808.07730. [1808.07730](#). (Cited on page [47](#).)
- Cameron, S.A., Eggers, H.C. and Kroon, S. (2019a). A sequential marginal likelihood approximation using stochastic gradients. *Proceedings*, vol. 33, no. 1. ISSN 2504-3900.  
Available at: <https://www.mdpi.com/2504-3900/33/1/18> (Cited on pages [6](#), [54](#), [60](#), [62](#), and [66](#).)

- Cameron, S.A., Eggers, H.C. and Kroon, S. (2019*b*). Stochastic gradient annealed importance sampling for efficient online marginal likelihood estimation. *Entropy*, vol. 21, no. 11. ISSN 1099-4300.  
Available at: <https://www.mdpi.com/1099-4300/21/11/1109> (Cited on pages 6, 8, 54, 60, 62, and 69.)
- Chao, M.T. (1982 12). A general purpose unequal probability sampling plan. *Biometrika*, vol. 69, no. 3, pp. 653–656. ISSN 0006-3444. <http://oup.prod.sis.lan/biomet/article-pdf/69/3/653/591311/69-3-653.pdf>.  
Available at: <https://doi.org/10.1093/biomet/69.3.653> (Cited on page 61.)
- Chen, T., B. Fox, E. and Guestrin, C. (2014 02). Stochastic gradient Hamiltonian Monte Carlo. *31st International Conference on Machine Learning, ICML 2014*, vol. 5. (Cited on pages 19, 23, 24, 27, 28, 29, and 63.)
- Cremer, C., Li, X. and Duvenaud, D. (2018 Jan). Inference Suboptimality in Variational Autoencoders. *arXiv e-prints*, p. arXiv:1801.03558. 1801.03558. (Cited on page 37.)
- C  rou, F. and Guyader, A. (2007). Adaptive multilevel splitting for rare event analysis. *Stochastic Analysis and Applications*, vol. 25, no. 2, pp. 417–443. <https://doi.org/10.1080/07362990601139628>.  
Available at: <https://doi.org/10.1080/07362990601139628> (Cited on page 32.)
- Doucet, A., de Freitas, N. and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science, 1st edn. Springer-Verlag New York. ISBN 978-1-4419-2887-0, 978-1-4757-3437-9. (Cited on page 50.)
- Girolami, M. and Calderhead, B. (2011). Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 2, pp. 123–214. <https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9868.2010.00765.x>.  
Available at: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2010.00765.x> (Cited on page 18.)

- Gorban, A.N., Sargsyan, H.P. and Wahab, H.A. (2010 Dec). Quasichemical Models of Multicomponent Nonlinear Diffusion. *arXiv e-prints*, p. arXiv:1012.2908. [1012.2908](#). (Cited on page [20](#).)
- Grosse, R.B., Ghahramani, Z. and Adams, R.P. (2015 Nov). Sandwiching the marginal likelihood using bidirectional Monte Carlo. *arXiv e-prints*, p. arXiv:1511.02543. [1511.02543](#). (Cited on pages [36](#), [37](#), [63](#), and [80](#).)
- Kampen, N.V. (2007). *Stochastic Processes in Physics and Chemistry (Third Edition)*. North-Holland Personal Library, third edition edn. Elsevier, Amsterdam. (Cited on page [19](#).)
- Kong, A., Liu, J.S. and Wong, W.H. (1994). Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 278–288. ISSN 01621459. Available at: <http://www.jstor.org/stable/2291224> (Cited on page [47](#).)
- Linden, W.v.d., Dose, V. and Toussaint, U.v. (2014). *Bayesian Probability Theory: Applications in the Physical Sciences*. Cambridge University Press. (Cited on pages [3](#), [40](#), and [41](#).)
- Ma, Y.-A., Chen, T. and Fox, E.B. (2015 Jun). A Complete Recipe for Stochastic Gradient MCMC. *arXiv e-prints*, p. arXiv:1506.04696. [1506.04696](#). (Cited on pages [28](#) and [29](#).)
- MacKay, D.J.C. (2002). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA. ISBN 0521642981. (Cited on page [31](#).)
- Mackevicius, V. (2013). *Introduction to Stochastic Analysis*. John Wiley & Sons, Ltd. ISBN 9781118603338. Available at: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118603338> (Cited on pages [19](#), [22](#), and [25](#).)
- Naesseth, C.A., Linderman, S.W., Ranganath, R. and Blei, D.M. (2017 May). Variational Sequential Monte Carlo. *arXiv e-prints*, p. arXiv:1705.11140. [1705.11140](#). (Cited on pages [51](#) and [81](#).)

- Naesseth, C.A., Lindsten, F. and Schön, T.B. (2019 Mar). Elements of Sequential Monte Carlo. *arXiv e-prints*, p. arXiv:1903.04797. [1903.04797](https://arxiv.org/abs/1903.04797). (Cited on pages [50](#) and [53](#).)
- Neal, R.M. (1998 Mar). Annealed Importance Sampling. *arXiv e-prints*, p. physics/9803008. [physics/9803008](https://arxiv.org/abs/physics/9803008). (Cited on pages [45](#) and [46](#).)
- Neal, R.M. (2008). The harmonic mean of the likelihood: Worst Monte Carlo method ever. <https://radfordneal.wordpress.com/2008/08/17/the-harmonic-mean-of-the-likelihood-worst-monte-carlo-method-ever/>. (Cited on page [35](#).)
- Robbins, H. and Monro, S. (1951 09). A stochastic approximation method. *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407. Available at: <https://doi.org/10.1214/aoms/1177729586> (Cited on page [19](#).)
- Rudin, W. (1987). *Real and Complex Analysis*. Mathematics series. McGraw-Hill. ISBN 9780071002769. Available at: <https://books.google.co.za/books?id=NmW7QgAACAAJ> (Cited on pages [21](#) and [38](#).)
- SKA (2019). Science data processor. Accessed: 2019-12-11. Available at: <https://www.skatelescope.org/signal-processing/> (Cited on page [4](#).)
- Skilling, J. (2004). Nested sampling. *AIP Conference Proceedings*, vol. 735, no. 1, pp. 395–405. <https://aip.scitation.org/doi/pdf/10.1063/1.1835238>. Available at: <https://aip.scitation.org/doi/abs/10.1063/1.1835238> (Cited on page [37](#).)
- Skilling, J. (2006 12). Nested sampling for general Bayesian computation. *Bayesian Anal.*, vol. 1, no. 4, pp. 833–859. Available at: <https://doi.org/10.1214/06-BA127> (Cited on pages [37](#) and [41](#).)
- Skilling, J. (2012). Bayesian computation in big spaces—nested sampling and galilean monte carlo. *AIP Conference Proceedings*, vol. 1443, no. 1, pp. 145–156. <https://aip.scitation.org/doi/pdf/10.1063/1.3703630>.

Available at: <https://aip.scitation.org/doi/abs/10.1063/1.3703630>  
(Cited on page 43.)

Springenberg, J.T., Klein, A., Falkner, S. and Hutter, F. (2016). Bayesian optimization with robust bayesian neural networks. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I. and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4134–4142. Curran Associates, Inc. Available at: <http://papers.nips.cc/paper/6117-bayesian-optimization-with-robust-bayesian-neural-networks.pdf> (Cited on page 29.)

Sutskever, I., Martens, J., Dahl, G. and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–1139–III–1147. JMLR.org. Available at: <http://dl.acm.org/citation.cfm?id=3042817.3043064> (Cited on page 27.)

Welling, M. and Teh, Y.W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In: Getoor, L. and Scheffer, T. (eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 681–688. Omnipress. Available at: [https://icml.cc/2011/papers/398\\_icmlpaper.pdf](https://icml.cc/2011/papers/398_icmlpaper.pdf) (Cited on page 26.)